NASA TECHNICAL NOTE

NASA TN D-3577



# A DIGITAL HIGHER ORDER INTERPOLATION PATH CONTROLLER

*by William Granville Batte*

*Langley Research Center*
*Langley Station, Hampton, Va.*



NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • SEPTEMBER 1966

# A DIGITAL HIGHER ORDER INTERPOLATION PATH CONTROLLER

By William Granville Batte

Langley Research Center
Langley Station, Hampton, Va.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – Price $2.50

CONTENTS

# A DIGITAL HIGHER ORDER INTERPOLATION PATH CONTROLLER*

By William Granville Batte
Langley Research Center

## SUMMARY

A two-axis digital higher order interpolation path controller for generating a smooth incremental path through discrete Cartesian input data is designed, constructed, and evaluated.

The novel interpolation technique exhibits the following features:

(1) Its digital implementation is simple relative to classical higher order interpolation schemes.

(2) It is readily adaptable to incremental techniques and to the calculus of finite differences.

(3) It accommodates unequal argument spacing.

(4) It processes multivalued and closed contour functions.

(5) It can accommodate raw or nonpreprocessed data.

(6) It is readily expandable to multidimensions.

Although the controller may be applied in many areas including control of machine tools, navigation, simulation, function generation, remote control, and automatic plotting, the experimental model is evaluated with a standard incremental x-y plotter as the output device and a punched paper tape reader as the input device.

Laboratory tests were made on the implemented system and actual copies of its output are included. These results show that the approach is feasible and that the research objectives are met.

## INTRODUCTION

The subject controller is a two-axis digital higher order interpolation device which generates a smooth incremental path through discrete, plane Cartesian input data. These data, which may fall into any one or more of the four quadrants, are sequentially ordered

---

*The information presented herein was included in a thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, Case Institute of Technology, Cleveland, Ohio, 1965.

as received so that the generated path follows this ordering. Since no attempt is made to fit the data to some analytic function, the controller may be viewed as a device for automatically "applying the draftman's french curve."

Although the controller is developed for use in automatic plotting, it may be applied in many other areas including control of machine tools, navigation, simulation, function generation, and remote control. Also, the mathematical scheme may be used as an interpolation technique for certain types of general purpose computing.

The controller is the result of research directed toward satisfying a need in automatic digital plotting. Automatic x-y plotters have replaced manual methods in most facilities where large quantities of data are displayed graphically. Most of these plotters produce point plots in which a pen or symbol head marks the locations corresponding to the various discrete x-y input values. In addition, some plotters (e.g., the Beckman 210 tape-to-plotter system) can also produce continuous curves in which the discrete input data are connected by straight-line segments. Not one plotter, however, is effective in "placing the french curve through the points" as was commonly done in the manual methods. Consequently the primary objective of this research is to evolve a simple controller capable of supplying the "french curve" even under conditions of (a) data with unequal argument spacing, (b) multivalued or closed contour functions, and (c) raw or nonpreprocessed data. Secondary objectives require the hardware to be relatively simple and fast enough to keep the output device operating at its inherent maximum speed.

Numerous digital controllers are in existence but each fails in one or more ways to satisfy these objectives. For example, Mergler's machine tool controller (ref. 1) requires rate information to be supplied as a part of the input data and, hence, does not satisfy condition (c). His proposed quadratic (second order) technique recognizes its inadequacies for certain "geometry" and, in addition, requires "equal intervals of the argument," each of which renders the scheme inadequate for the present application.

Several second-order systems are in existence but they too do not satisfy condition (c). For example, the Bendix Dynapath system (ref. 2) requires, for circular interpolation, that the initial and final positions of the radius vectors be supplied as a portion of the input data. Similarly, the Fuji system (ref. 3) requires that the input data be expressed in a coordinate system whose origin is at the center of the arc. Even more serious, however, is that the present application requires at least a third-order interpolation as demonstrated in reference 4.

Ninke (ref. 5) recently adapted the Newton-Gauss interpolation technique to a digital third-order controller. Using digital differential analyzer (DDA) methods, the system consists of three digital integrators cascaded so that the third derivative introduced at the first integrator is successively integrated three times, this integration yielding the path function at the output of the last integrator. The initial conditions of the integrators

and corrections thereto must be supplied as a portion of the input data; hence, condition (c) is violated. Also the simplicity with which these quantities are computed is a result of the assumption by Ninke that the input data are spaced at equal increments of the arguments; hence, condition (a) is also violated.

A similar scheme is outlined in reference 6 where again the approach is based on equal arguments.

As with most digital systems the controller has its analog counterpart. In many situations, however, the accuracy requirement renders the analog approach ineffective. Consider, for example, even simple linear interpolation where it is required to connect two points with a straight line. This problem is approached with the conventional analog closed-loop plotter by integrating the inputs to the two axes with the same effective time constants. The solution sounds simple, but the word "effective" requires a consideration of the unbalanced inertias and frictions on the two axes, the unbalanced breakaway torques, and, of course, settings of gain, damping, and so forth. Furthermore, higher-order interpolation is even more complex!

The study of existing controllers and classical interpolation techniques as applied to the performance requirements of the subject controller leads to the following broad observations which are presented in the nature of hypotheses upon which the controller's design is based:

(1) It is not important to fit the input data to some analytic function (e.g., a third-degree polynomial or perhaps some french curve logarithmic function); instead, it is more important to utilize some hardware-oriented (though perhaps less mathematical) scheme such as a goal seeking system.

(2) Since the entire path must be generated, as opposed to merely evaluating a few isolated function values, a finite difference technique should be utilized such that the function change is computed and used to update the function along the generated path.

(3) Since the finite difference technique will no doubt result in many isolated arithmetic units, the internal coding should be natural binary with negative numbers represented in two's complement notations. This coding will simplify the designs of the units.

(4) Since any closed contour path contains at least two infinite slopes, a technique which obviates the large slope problem (refs. 5 and 6) must be used.

(5) For simplicity of the experimental hardware, an incremental, open-loop control system should be used. (Where applications require absolute or incremental closed-loop systems, the experimental design is easily adapted to them.)

(6) Since the path must be generated from any data point through the next point, the direction of the latter point with respect to the present position of the generated path is an important variable and should probably appear explicitly in the basic interpolation scheme.

Other philosophies peculiar to various parts of the controller are found in the appropriate design sections.

## SYMBOLS AND NOTATIONS

### Literals

Arithmetic:

| | |
|---|---|
| a,b,c,d,e | set of input data points |
| AJ | accumulator J where J is any integer |
| (AJ) | quantity in AJ |
| (AJx),(AJy) | quantity in x or y portion of AJ |
| CJ | counter J where J is any integer |
| (CJ) | quantity in CJ |
| $d\psi$,dx,. . . | infinitesimal differentials |
| $\frac{d\phi}{ds}$ | instantaneous curvature of a trajectory |
| F | multiplier associated with binary rate multiplier |
| K | constant which relates two equivalent angles $\psi_1$ and $\psi_2$ |
| k,k-1,k+1 | present, previous, and next values of a discrete quantity |
| L | length of (i.e., number of bit positions or stages in) accumulator, counter, or register |
| p | point on generated path |
| Q | proportionality constant which determines trajectory |
| R | carriage return character on tape |
| RJ | register J where J is any alphabetical character |
| (RJx),(RJy) | quantity in x or y portion of RJ |
| s | trajectory path |

| | |
|---|---|
| S[ ] | sign of quantity within brackets |
| SWJ | switch J where J is any integer or alphabetical character |
| x,y,z | Cartesian coordinates |
| $\alpha$ | tangent angle of generated path at last data point |
| $\beta$ | tangent angle of generated path at next data point |
| $\gamma$ | corrected error distance between grid corner and s-curve—grid-line intersection |
| $\delta(C4_c)$ | change in coarse portion of quantity in counter 4 |
| $\delta\psi,\delta x,\ldots$ | finite difference or change in $\psi$, x, ... |
| $\Delta$ | "space" character on tape separating x- and y-data |
| $\Delta\psi,\Delta\phi,\ldots$ | total change in $\psi$, $\phi$, ... |
| $\epsilon$ | error between $\phi$ and $\theta_c$ |
| $\theta$ | motion angle required for coincidence with s-curve at next grid crossing; subscript c or f denotes coarse or fine portion of $\theta$, respectively |
| $\theta_{c\phi}$ | merging function which starts from $\theta_c$ and ends at $\phi$ |
| $\rho$ | predicted error distance between grid corner and s-curve—grid-line intersection |
| $\phi$ | steering angle or instantaneous angle of tangent to theoretical trajectory |
| $\psi$ | angle of present position with respect to next data point |
| \|CJ\| | absolute value of (CJ) |

Prime with symbol indicates quantity is expressed in transformed coordinate system.

Logic:

| | |
|---|---|
| A,B,C,. . . | Boolean variables |
| $A_k$,$A_{k+1}$ | augend (minuend) bit in accumulator before and after arithmetic operation, respectively |
| $B_k$ | addend (subtrahend) bit input to accumulator |
| A[AJ],B[AJ] | A and B inputs of AJ |
| N | accumulator (counter) is subtracting (counting backwards) |
| N[AJ] | operation of AJ is negative |
| $P_k$,$P_{k+1}$ | carry (borrow) bit in accumulator before and after addition (subtraction), respectively |
| P[AJ] | propagation signal P of AJ |
| SPGJ | shift pulse generator J where J is any integer |
| $S_J$,$R_J$,$T_J$ | set, reset, and trigger (complementary) inputs to flip-flop J where J is any alphabetical character |
| S[ ] | sign of quantity within brackets is negative |
| X | "don't care" condition |
| $\alpha_J$ | flip-flop J is setting (i.e., is undergoing "0" to "1" transition) or logical signal J is turning on (J is any alphabetical character) |
| $\alpha_{\underline{J}}^{n\tau}$ | defined as $\alpha_J$ except signal is delayed by $n\tau$ units and J is in terms of positive logic |
| $\beta$ | angle $\beta$ is being computed; $\overline{\beta}$ implies that angle $\psi$ is being computed |
| $\beta_J$ | flip-flop J is resetting (i.e., is undergoing "1" to "0" transition) or logical signal J is turning off (J is any alphabetical character) |

$|\delta x|, |\delta y|$ 1 increment of motion is occurring in x- or y-direction (without regard to its sense)

$\tau$ unit of delay

A bar over a symbol indicates the negation (or complement) of a variable.

An underscore with a Boolean variable designates the positive logic convention; the absence of the underscore represents the more common negative logic convention.

Logic Design

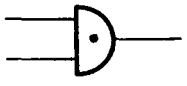
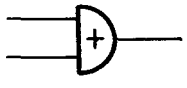
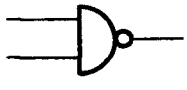
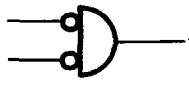
. AND (conjunction or intersection)
(The dot may be omitted.)

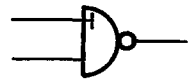
+ (INCLUSIVE-)OR (disjunction or union)

$\oplus$ EXCLUSIVE-OR

$\Rightarrow$ implication

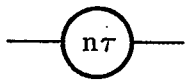
$\subset$ inclusion

| | | Boolean characterization (upper left line is input signal A; lower left line is input signal B; right line is output signal C) |
|---|---|---|
|  | AND | $C = AB$ |
|  | OR | $C = A + B$ |
|  | NAND serving as AND | $\underline{C} = AB$ |
|  | NAND serving as OR | $C = \underline{A} + \underline{B}$ |

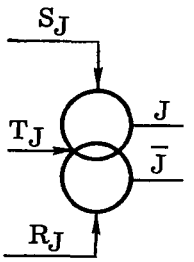


gated pulse generator $\underline{C} = \alpha_A B$

steering gate $\underline{C} = \underline{A}\alpha_{\underline{B}}$




pulse delayer $C = \alpha_A^{n\tau}$ (No B input)




flip-flop J where $S_J$, $R_J$, $T_J$ are set, reset, and trigger inputs and J, $\bar{J}$ are assertion and negation outputs, respectively

## GENERAL THEORETICAL CONSIDERATIONS

In generating a curve from any input point a to the next point b with slope $\beta$ at b (fig. 1), $\psi$ (the angle of b with respect to the presently generated position p) may be determined and easily transformed to a new (primed) coordinate system (fig. 2) whose origin is at b and whose X-axis is alined with $\beta$. The tangent angle $\phi'$ may then be determined from the simple relation $\phi' = 2\psi'$ and easily transformed back to the original coordinates to guide the curve generation of one increment. The process may be iterated until point b is reached. (See appendix.)

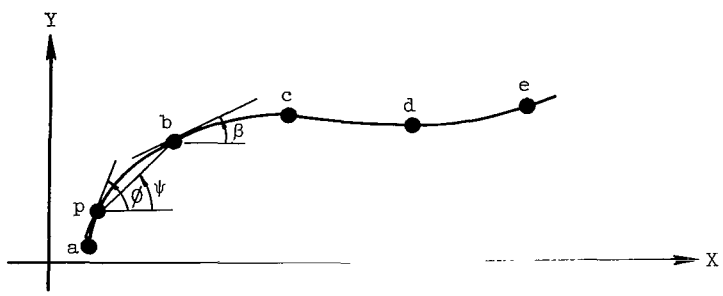



Figure 1.- Interpolated curve through input data points a, b, c, d, e.

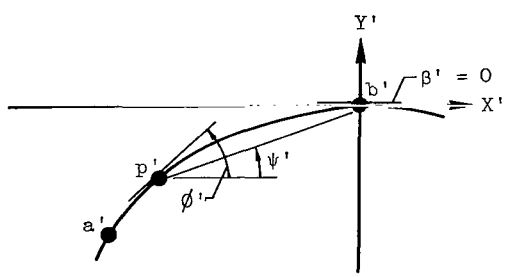



Figure 2.- Portion of figure 1 transformed to new coordinate system.

The relation $\phi' = 2\psi'$ is a special case of

$$\phi' = Q\psi' \qquad (1)$$

which, for appropriate values of the parameter Q, is the equation of well-known geometric curves which go through the coordinate's origin at zero slope; for example, Q values of 1.5, 2.0, and 3.0 yield cardioids, circles, and lemniscates, respectively, where the size of these curves is determined by the initial point a.

The subject controller's operation, which is based on the foregoing principles, is outlined and illustrated by an example in the next section.

## Computational Algorithms

Point p of figure 1 is moved in finite steps from a to b along a path ending with slope $\beta$ by use of the following algorithm:

(1) Compute the direction angle $\psi$ of the vector pb.

(2) Transform $\psi$ into $\psi'$, expressed in a new coordinate system centered at b and alined with $\beta$.

(3) Compute $\phi' = Q\psi'$.

(4) Inversely transform $\phi'$ into $\phi$, expressed in the original coordinate system.

(5) Use $\phi$ to predict the next increment of the curve, thereby establishing a new p.

(6) Repeat steps (1) through (5) until p and b coincide.

A step-by-step example of this simplified algorithm, in which it is assumed that (a) the coordinates of points a and b are, respectively, $x_a = 3$, $y_a = 2$ and $x_b = 10$, $y_b = 12$, (b) for $x_a \leq x_p < x_b$, $\beta = 20^o$ (for the computation of $\beta$, see p. 20), and (c) the desired curve is circular and hence $Q = 2$, is as follows:

(1) $\psi = \tan^{-1}\left(\frac{y_p - y_b}{x_p - x_b}\right)$

$= \tan^{-1}\left(\frac{2 - 12}{3 - 10}\right) = 55^o$ (since initially point p coincides with point a)

(2) $\psi' = \psi - \beta$ (appendix)

$= 55^o - 20^o = 35^o$

(3) $\phi' = Q\psi'$

$= 2(35^o) = 70^o$

(4) $\phi = \phi' + \beta$

$= 70^o + 20^o = 90^o$

(5) Since $\phi = 90^o$, "increment" p one unit in the plus y-direction.

(6) Since the new p does not yet coincide with b, return to step (1).

(1) $\psi = \tan^{-1}\left(\frac{3 - 12}{3 - 10}\right) = 52^{o}$

(2) $\psi' = 52^{o} - 20^{o} = 32^{o}$

(3) $\phi' = 2(32^{o}) = 64^{o}$

(4) $\phi = 64^{o} + 20^{o} = 84^{o}$

(5) Since the motion of an incremental orthogonal two-axis system is restricted to some multiple of $45^{o}$ and since $90^{o}$ is the closest multiple to $84^{o}$, again increment p one unit in the plus y-direction. However, simply to ignore this $6^{o}$ discrepancy and similar ones for the various increments of the path would result in the generation of an octagonal shape instead of the desired circular one. The problem is solved by integrating these discrepancies in accordance with the method developed in the section "Derivation of the Output Function" and summarized on page 17.

(6) Return to step (1).

The foregoing procedure is repeated until p and b coincide, at which time the entire process is repeated for the next segment[1] of the curve. In general, there will be a slope discontinuity at the beginning of each new segment of the curve; this condition is rectified by the method presented in the section "Merging Function Criteria."

The algorithm is now refined with regard to the computation of $\psi$. The relation $\psi = \tan^{-1}(f(x,y))$ is relatively difficult for machines to compute as seen in reference 4. Unfortunately, step (1) requires this computation for each increment of the curve. Alternatively, however, the algorithm may be changed to utilize, instead, the more easily computed $\delta\psi$, the finite change in $\psi$. The refined algorithm is as follows:

(1) Compute the direction angle $\psi$ of the vector pb.

(2) Transform $\psi$ into $\psi'$.

(3) Compute $\phi' = Q\psi'$.

(4) Transform $\phi'$ into $\phi$.

(5) Increment p in accordance with $\phi$.

(6) Compute $\delta\psi$ due to the movement of p.

(7) Increment $\phi$ by $Q\,\delta\psi$.

(8) Repeat steps (5) through (8) until p and b coincide.

---

[1] The interpolated curve is composed of segments connecting adjacent data points.

This algorithm is advantageous in that the slow computations of steps (1) through (4) are performed once, with only the fast operations of steps (5) through (8) iterated for each increment of the curve.

Step (7) is based on the difference equation

$$\phi_{k+1} = \phi_k + Q\ \delta\psi_k$$

Since, by the fundamental difference definition

$$\phi_{k+1} = \phi_k + \delta\phi_k$$

it is necessary to show only that

$$\delta\phi_k = Q\ \delta\psi_k$$

By the invariance property of the linear transformation (appendix),

$$\delta\phi_k = \delta\phi'_k$$

By the fundamental definition,

$$\delta\phi_k = \phi'_{k+1} - \phi'_k$$

From equation (1),

$$\delta\phi_k = Q\psi'_{k+1} - Q\psi'_k$$

Again, by the fundamental definition,

$$\delta\phi_k = Q\ \delta\psi'_k$$

Finally, by the invariance property,

$$\delta\phi_k = Q\ \delta\psi_k$$

## Derivation of Output Function

Step (5) of the refined algorithm requires that p be incremented "in accordance with $\phi$." The coarse $45^o$ angular resolution inherent in the output of an incremental, orthogonal, two-axis system poses a problem, in this instance, since such a system must generate $\phi$, a function of considerably finer resolution.

The problem is first approached in a precise manner. From this approach a complicated partial solution is obtained. Then applied is an approximation which yields a complete and greatly simplified solution without, for most applications, excessively degrading the results.

Figure 3 shows a magnified portion of the output grid on an incremental device, function, and so forth. Intersections of grid lines represent stable or valid output locations, quantities, and so forth. Curve s represents the desired, computed, interpolated path or function where $\phi$ is the angle of the vector or directional tangent[2] to the curve. For purposes of this discussion, the grid spacing is considered small enough with respect to the curvature of the path $d\phi/ds$ to assume $\phi$ to be constant over the two increments shown in figure 3. Other parameters in figure 3 are defined as follows:

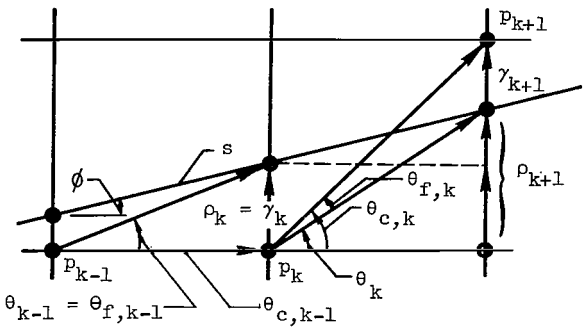



Figure 3.- Magnified portion of incremental output grid.

$\theta$ desired motion angle necessary for coincidence with s at its next grid crossing

$\theta_c$ actual motion angle

$\theta_f$ difference angle defined implicitly by

$$\theta = \theta_c + \theta_f \tag{2}$$

where $\theta_c$ and $\theta_f$ may be construed as coarse and fine components of $\theta$ and are subject to the two following constraints:

$$0 \leqq \left[\theta_c \equiv 0 (\mathrm{mod}\ 45^o)\right] < 360^o$$

and

$$-22.5^o \leqq \theta_f < 22.5^o$$

$\rho,\gamma$ predicted and corrected error distances, respectively, between grid corners and s-curve—grid-line intersection

k-1,k,k+1 subscripts denoting order of the motion increments

---

[2]The tangent is not the usual scalar tangent since the direction or sense in which the interpolated curve is generated affects $\phi$ – that is, if the sense of generation is reversed, then $\phi$ is rotated by $180^o$.

The solution proceeds as follows: $\rho_{k+1}$, the error distance which would result if the motion continued in the direction of $\theta_{c,k-1}$, is calculated. If $|\rho_{k+1}| \leqq 0.5$ increment, then

$$\theta_{c,k} = \theta_{c,k-1}$$

and

$$\gamma_{k+1} = \rho_{k+1}$$

If $|\rho_{k+1}| > 0.5$ increment, then

$$\theta_{c,k} = \theta_{c,k-1} \pm 45^{o}$$

and

$$\gamma_{k+1} = \rho_{k+1} \pm 1$$

With the aid of figure 3, the procedure may be formulated for $-45^{o} \leqq \theta_{c,k-1} \leqq 45^{o}$ as

$$\rho_{k+1} = \gamma_{k} + \tan \phi_{k} - \tan \theta_{c,k-1} \tag{3}$$

$$\gamma_{k+1} = \rho_{k+1} - I$$

$$\theta_{c,k} = \theta_{c,k-1} + I$$

where

$$I = \frac{\rho_{k+1}}{|\rho_{k+1}|} I\left[|\rho_{k+1}| + 0.5\right]$$

I[ ] is defined as the greatest integer in quantity within the brackets, and $\tan \theta_{c,k-1}$ is necessary, when $\theta_{c,k-1} = \pm 45^{o}$.

The foregoing formulation will be used in the derivation of equation (7); first, however, it is necessary to establish the relationship between $\rho$ and $\theta_{f}$. Figure 3 shows that $\rho$ may be expressed also as

$$\rho_{k+1} = \tan \theta_{k} = \tan\left(\theta_{c,k} + \theta_{f,k}\right) \tag{4}$$

$$\rho_{k+1} = \frac{\tan \theta_{c,k} + \tan \theta_{f,k}}{1 - \tan \theta_{c,k} \tan \theta_{f,k}} \tag{5}$$

For $\theta_{c,k} = 0$, equation (4) reduces to $\rho_{k+1} = \tan \theta_{f,k}$. Hence, a change in $\theta_{c,k}$ should take place when

$$\left|\theta_{f,k} = \tan^{-1}(\rho_{k+1})\right| > \left(\left|\tan^{-1}(\pm 0.5)\right| = 26.6^{o}\right)$$

For $\theta_{c,k} = +45^{o}$, equation (5) reduces to

$$\rho_{k+1} = \frac{1 + \tan \theta_{f,k}}{1 - \tan \theta_{f,k}} = \frac{\cos \theta_{f,k} + \sin \theta_{f,k}}{\cos \theta_{f,k} - \sin \theta_{f,k}}$$

$$= \frac{\cos^2 \theta_{f,k} + 2 \cos \theta_{f,k} \sin \theta_{f,k} + \sin^2 \theta_{f,k}}{\cos^2 \theta_{f,k} - \sin^2 \theta_{f,k}}$$

$$= \frac{1 + 2 \cos \theta_{f,k} \sin \theta_{f,k}}{1 - 2 \sin^2 \theta_{f,k}} = \frac{1 + \sin 2\theta_{f,k}}{\cos 2\theta_{f,k}}$$

$$\rho_{k+1} = \sec 2\theta_{f,k} + \tan 2\theta_{f,k} \tag{6}$$

Although the inverse $\theta_{f,k}(\rho_{k+1})$ is more complicated, it is easily seen by substitution that a change in $\theta_{c,k}$ should occur when

$$\left|\theta_{f,k}\right| > (45^{o} - 26.6^{o} = 18.4^{o})$$

that is, when $\theta_{f,k} < -18.4^{o}$.

Finally, for $\theta_{c,k} = -45^{o}$, equation (5) reduces to

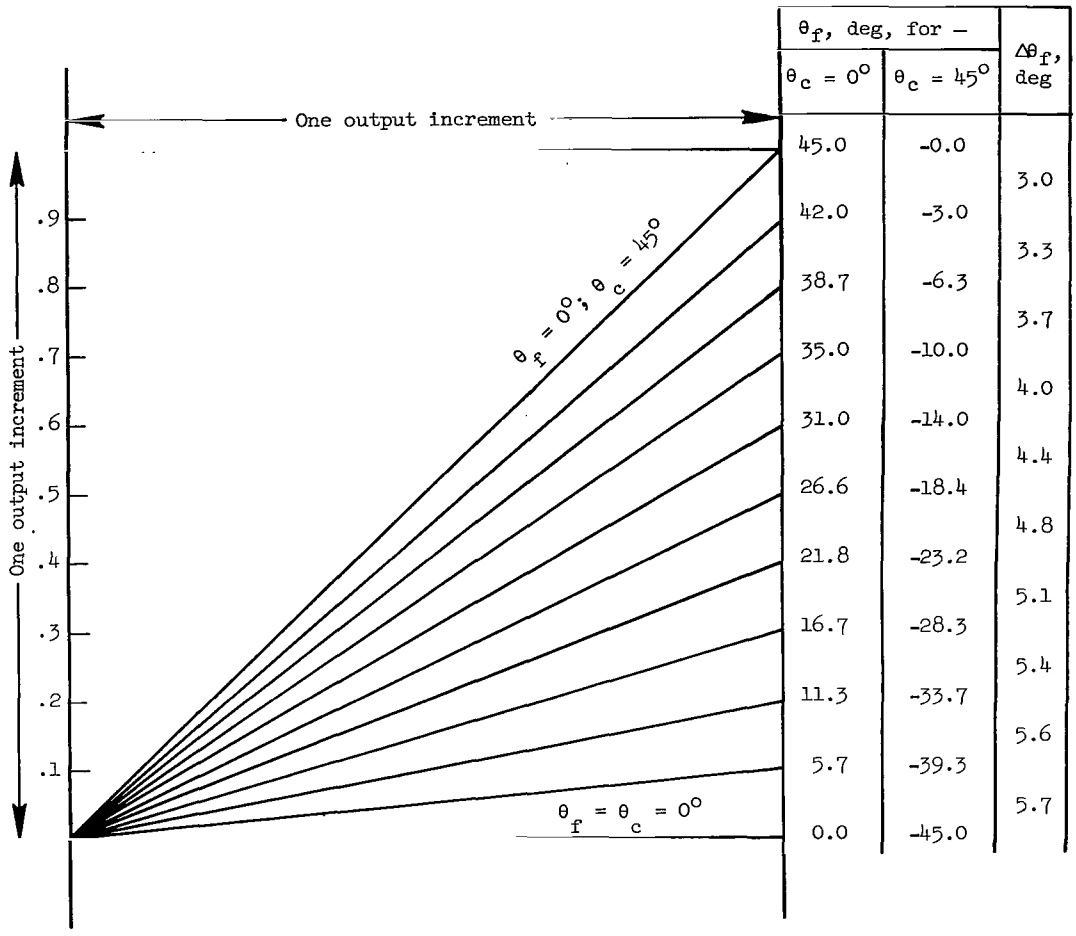
$$\rho_{k+1} = \frac{-1 + \tan \theta_{f,k}}{1 + \tan \theta_{f,k}}$$

Following the foregoing procedure results in

$$\rho_{k+1} = \tan 2\theta_{f,k} - \sec 2\theta_{f,k}$$

and, hence, a change in $\theta_{c,k}$ should occur when $\theta_{f,k} > +18.4^{o}$.

Even considering only these limited cases, the $\rho_{k+1}$ function is relatively complicated. To assist in the application of a reasonable approximation, the function is displayed, with the aid of figure 4, in figure 5. It is noted that to implement $\rho$ accurately requires the use of one complicated function if $\theta_{c} = 0^{o}$ and another complicated function




| $\theta_f$, deg, for – | | $\Delta\theta_f$, deg |
|---|---|---|
| $\theta_c = 0^o$ | $\theta_c = 45^o$ | |
| 45.0 | -0.0 | |
| | | 3.0 |
| 42.0 | -3.0 | |
| | | 3.3 |
| 38.7 | -6.3 | |
| | | 3.7 |
| 35.0 | -10.0 | |
| | | 4.0 |
| 31.0 | -14.0 | |
| | | 4.4 |
| 26.6 | -18.4 | |
| | | 4.8 |
| 21.8 | -23.2 | |
| | | 5.1 |
| 16.7 | -28.3 | |
| | | 5.4 |
| 11.3 | -33.7 | |
| | | 5.6 |
| 5.7 | -39.3 | |
| | | 5.7 |
| 0.0 | -45.0 | |

Figure 4.- Error angle $\theta_f$ for various error distances $\rho$.

if $\theta_c = \pm 45^o$. The approximation shown in figure 5, however, greatly simplifies the problem since

(1) It is linear and thus is simply $\rho = C\theta_f$ where C is the proportionality constant.

(2) The same function is applied for both conditions of $\theta_c$.

(3) Its linearity enables the application of the following superposition property for $\rho(\theta)$: $\rho(\theta_1) + \rho(\theta_2) = \rho(\theta_1 + \theta_2)$. Note further that C vanishes when $\theta_f$ is measured in octants (1 octant $\equiv 45^o$) and $\rho$ in increments – that is, $\rho = \theta_f$.

(4) It avoids the problem of large tangents and, therefore, can be used throughout the entire $360^o$.

The penalty for using this approximation is exhibited in the digital computer simulation of reference 4, appendix C, where the attempted difficult task of generating a circle through two points and one slope resulted in four sides slightly flattened. For normal

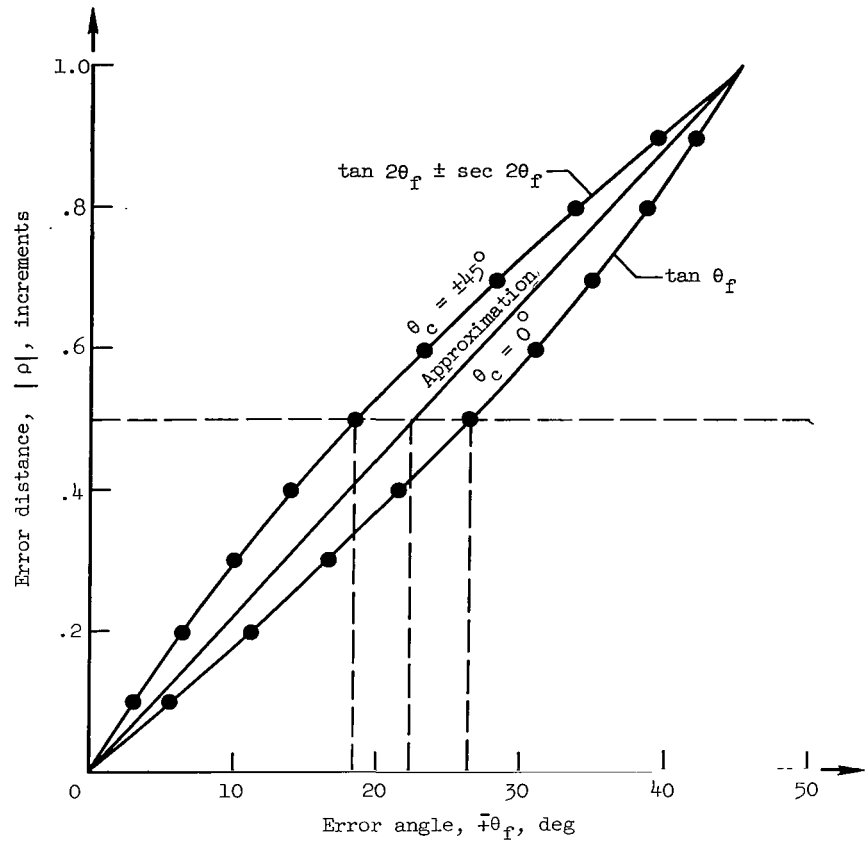



Figure 5.- Relation between actual and approximate error angle $\theta_f$ and error distance $\rho$.

curve interpolation tasks, however, this slight tendency to flatten is insignificant. The implementation, therefore, is based on the use of this approximation.

In effect, then, the approximation establishes an equivalence relation between angular dimensions at $p_k$ of figure 3 and linear dimensions at the $p_{k+1}$ increment crossing, so that the tangent functions are eliminated and thereby the useful angular domains are extended throughout the entire $360^o$. Implicit in the relationship $\rho = \theta_f$ is the corrected form $\gamma = \theta_f$ (where the constraints of equation (2) have been applied to the latter $\theta_f$). Hence equation (3) can be written as

$$\theta_{f,k} = \theta_{f,k-1} + \phi_k - \theta_{c,k-1} \tag{7}$$

where the constraints of equation (2) are not yet applied to $\theta_{f,k}$. By updating the subscripts

$$\theta_{f,k+1} = \theta_{f,k} + \phi_k - \theta_{c,k} \tag{8}$$

where, as assumed, $\phi_{k+1} = \phi_k$. Therefore,

$$\delta\theta_{f,k} = \theta_{f,k+1} - \theta_{f,k} = \phi_k - \theta_{c,k}$$

Integrating finitely (ref. 7) gives

$$\theta_{f,k} = \theta_{f,0} + \sum_{j=0}^{k-1} \delta\theta_{f,j}$$

where $j$ is a dummy variable. By letting the initial condition $\theta_{f,0} = 0$,

$$\theta_{f,k} = \sum_{j=0}^{k-1} \epsilon_j \tag{9}$$

where $\epsilon_j = \delta\theta_{f,j} = \phi_j - \theta_{c,j}$, the error between the tangent angle of curve $s$ and the actual motion angle. Hence equation (2) can be written

$$\theta_k = \theta_{c,k} + \sum_{j=0}^{k-1} \epsilon_j$$

When $\sum \epsilon_j$ exceeds the constraints of equation (2), 1 or more octants are transferred from $\sum \epsilon_j$ to $\theta_c$. Equation (9) is a most important result since step (5) of the refined algorithm can now be stated explicitly:

(1) Determine $\theta_{f,k}$ for each point $p_k$ along the generated path by finitely integrating or summing the error $\epsilon_j$ from initial $\epsilon_0 = 0$ to $\epsilon_{k-1}$, continually applying the constraints associated with equation (2).

(2) Use equation (2) and associated constraints to establish $\theta_{c,k}$ for each point $p_k$.

(3) Generate the path in the direction of $\theta_{c,k}$.

Thus, $\epsilon_j$ as previously defined represents the error introduced at the jth step because the actual direction of travel $\theta_{c,j}$ differs from the computed direction $\phi_j$. Whenever $|\theta_{f,k}|$, the finite integral of this error integrated from an input data point $p_0$ to the previous generated point $p_{k-1}$, exceeds $22\frac{1}{2}^o$, it applies to $\theta_{c,k}$ a $45^o$ correction which, in general, is an overcorrection, in which case $\epsilon_j$ changes sign and $\theta_{f,k}$ builds up in the opposite sense. Hence the actual path (controlled by $\theta_c$) continually seeks and, in general, follows the computed path (controlled by $\phi$) to within 0.5 increment. In

terms of the CalComp digital incremental recorder model 565 that was used in this research, this 0.5 increment represents an error of 0.005 inch, a distance approximately as small as the plotted curve thickness.

The beauty of this formulation is that it is easily implemented.

## Merging Function Criterion

In generating a curve from point a to point b, the refined algorithm of page 10 gives no consideration to the angle at which the curve originally entered a. In general, disregard of this angle results in derivative discontinuities at the various input data points of a curve.

To obviate this problem, a merging function is derived which effects a smooth derivative transition from the curve entering some point to the curve leaving this point by limiting $d\phi/ds$, the curvature of the path in the vicinity of the point. The particular merging function criterion which is implemented may be considered to be circular since it limits $d\phi/ds$ to the $d\phi/ds$ of a circle whose diameter is equal to the distance between the present position p and the next data point. For the circle shown in figure 6 $d\phi/ds$ is constant over the path s; hence,

$$\frac{d\phi}{ds} = \frac{2\pi \text{ radians}}{\pi d_q \text{ increments}} = \frac{2}{d_q} \frac{\text{radians}}{\text{increment}} \leqq \frac{2}{d_p} \frac{\text{radians}}{\text{increment}} \tag{10}$$

If $d\phi/ds$ is limited to $2/d_q$ for an interpolated path, the merging function gives, in general, (1) a smooth transition, and (2) complete merging before p coincides with b.

Although other merging criteria could have been used, this particular one is attractive since it is a simple function of $d_p$, the square of which is already computed by the experimental hardware for other purposes.

The merging function is shown in figure 7 as the limiting $d\phi/ds$ plotted against $d_p$. Limiting $d\phi/ds$ is given in radians per increment and $\pi$ radians per increment where $1\pi$ radian = $\pi$ radians. The implementation described in the section "Operations of the Experimental Model" is based on an approximation to this function.

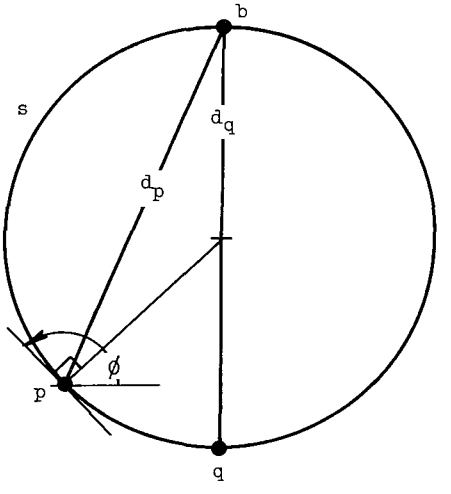



Figure 6.- Circular criterion for merging function.

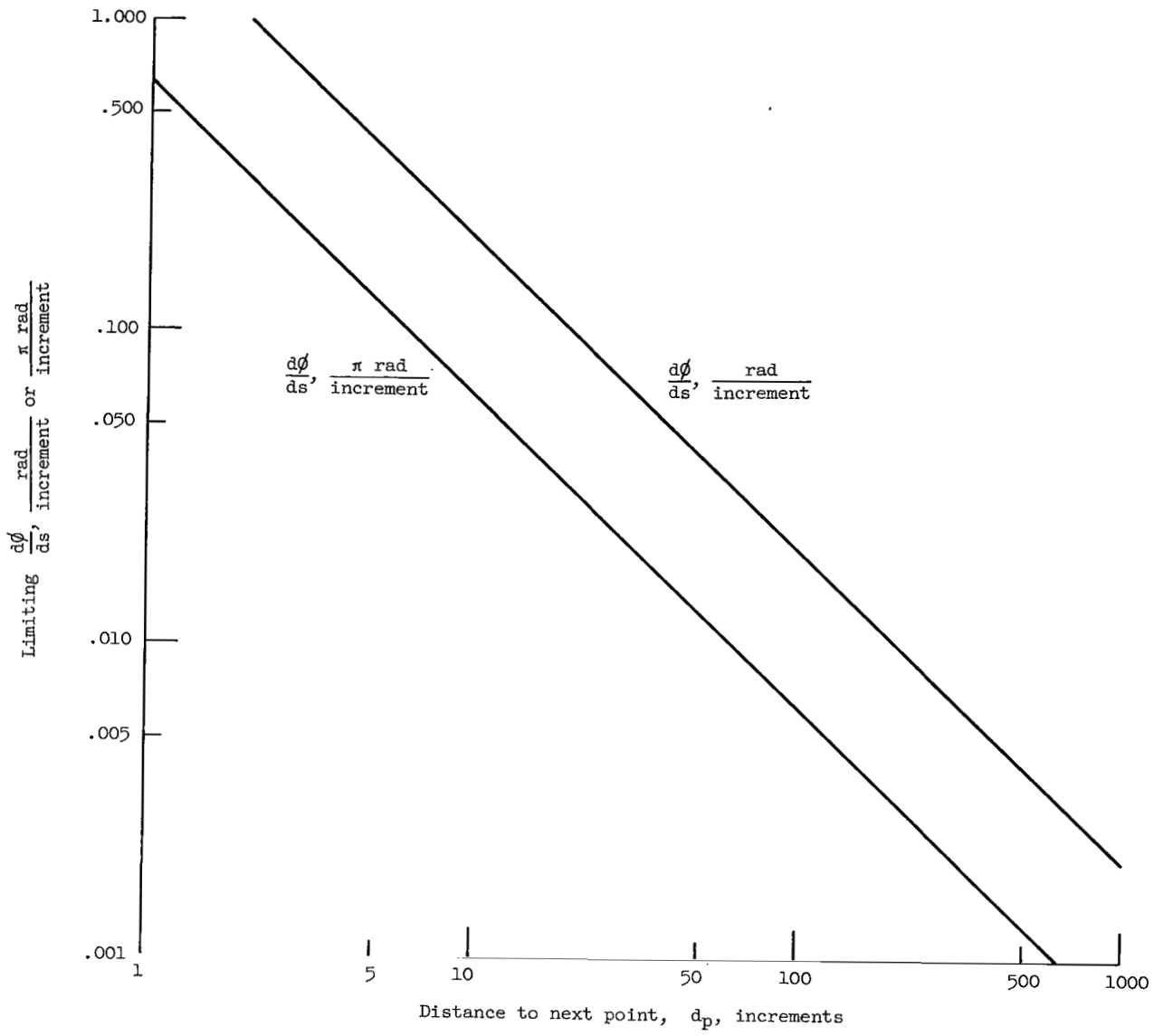



Figure 7.- Merging function.

## Computation of $\beta$

The development of the theory is based on the hypothesis that the slope angle $\beta$ of figure 1 is given. An examination of this hypothesis is now made.

In general, a unique accurate slope angle $\beta$ does not exist since the input points do not specify the curve in the immediate neighborhood of point b – that is, in digital terms, the input points do not specify the increments immediately preceding and following point b. Therefore, the slope assigned to point b is a function of the particular

interpolation scheme used. On the assumptions that sufficient input points are given to faithfully represent the function and that point b lies in the middle 50-percent region of the distance between a and c $\left(\text{or, more precisely, } \frac{1}{3} < \lambda < 3, \text{ where } \lambda \text{ is the ratio of the distance ab to the distance ac}\right)$, then the simple first difference definition of

$$\beta = \tan^{-1} \frac{y_a - y_c}{x_a - x_c}$$

is, in general, adequate as is seen by examining the geometry of various three-point patterns.

## Equivalence of Angles

Physically $\psi$, $\phi$, $\beta$, and so forth, can range only over $2\pi$ radians. Hence, physical significance is extended to those values outside of range by letting the equivalence relation

$$\psi_1 \equiv \psi_2$$

mean $\psi_1 = \psi_2 + 2\pi K$ for some integer K $\left(\text{i.e., } \psi_1 \equiv \psi_2 (\text{mod } 2\pi)\right)$.

## Principal Range of Angles

With the aid of the foregoing equivalence relation, $\psi'$ in any range may be converted to its principal range defined as

$$-\pi \leqq \psi' < +\pi$$

The important relation $\phi' = Q\psi'$ tacitly assumes that $\psi'$ is expressed in its principal range. However, it is shown in the following theorem that this requirement is really not a necessary condition when Q is an integer; that is, for the relation $\phi' = Q\psi'$ equivalent angles $\phi'$ are obtained for equivalent angles $\psi'$ when Q is an integer.

Theorem: Under the relations

$$\phi_1 = Q\psi_1$$

$$\phi_2 = Q\psi_2$$

$$\psi_2 \equiv \psi_1 (\text{mod } m) \text{ where } m \text{ is any real number,}$$

if Q is an integer then $\phi_2 \equiv \phi_1 (\text{mod } m)$.

Proof: (1) $\psi_2 \equiv \psi_1 (\text{mod } m)$, by hypothesis.

(2) $\psi_2 = \psi_1 + mK$, by the equivalence definition.

(3) $\phi_2 = Q\psi_2$, by hypothesis.

(4) $\phi_2 = Q(\psi_1 + mK)$, by steps (2) and (3).

(5) $\phi_2 = \phi_1 + mKQ$, by hypothesis.

(6) $Q$ is an integer, by hypothesis.

(7) $KQ$ is an integer, since the product of two integers is an integer.

(8) Therefore $\phi_2 \equiv \phi_1 (\text{mod } m)$, by steps (5) and (7) and the equivalence definition.

The converse is not true. However, an important fact concerning the converse is stated in the following corollary to the theorem.

Corollary: Under the relations of the theorem, the equivalence of $\phi_1$ and $\phi_2$ is guaranteed only if $Q$ is an integer.

This corollary is a consequence of step (7) of the theorem proof; that is, although specific cases can be found in which $KQ$ is an integer for a noninteger value of $Q$, an integer product is assured only if both $K$ and $Q$ are integers.

These considerations – angular equivalence, principal range, and integer values of $Q$ – are extremely important in terms of hardware implementation. Their significance will be briefly mentioned so that they can be more fully appreciated when the actual design of the experimental model is presented in a subsequent section.

The angle equivalence relation

$$\psi_2 \equiv \psi_1 (\text{mod } 2\pi)$$

implies that the hardware need not distinguish angles which differ by integral multiples of $2\pi$. Therefore the associated counters, registers, and so forth, are required to handle precisely the modulus $2\pi$ range and, hence, bit positions of weight $\geq 2\pi$ need not be provided even though the angle may exceed $2\pi$.

The theorem essentially states that the complication associated with the principal range constraint can be avoided if one is satisfied with interpolated curves involving only integer values of $Q$ – that is, $Q = 1$, the linear case, $Q = 2$, the circular case, and so forth. Utilizing only integer values of $Q$ would greatly simplify the system control problem and could eliminate the binary rate multiplier in the design of the experimental model. The importance of noninteger $Q$ in a practical operational device can best be established through operational experience. The more complicated noninteger $Q$ capability is therefore retained in the model design so that some measure of its importance may be acquired through use of the experimental device.

## COMPUTATION OF $\psi$

Perhaps the most important single function to be performed in the interpolation scheme is the computation of

$$\psi = \tan^{-1} \frac{\Delta y}{\Delta x}$$

or, if one of the points is at the origin as in figure 8,

$$\psi = \tan^{-1} \frac{y}{x}$$

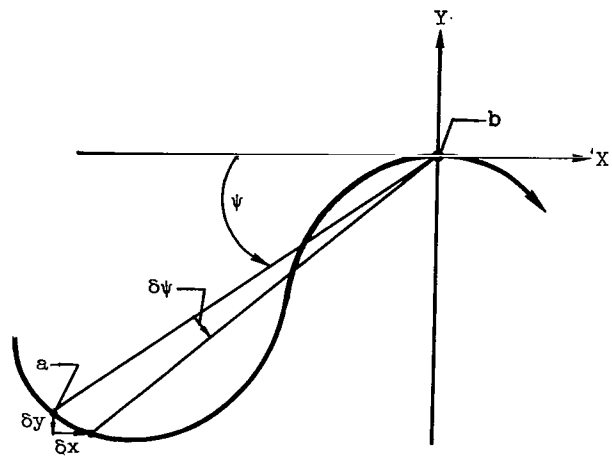



Figure 8.- Effect of $\delta x$ and $\delta y$ on $\psi$.

Thus, with a suitable arc tangent algorithm, $\psi$ can be computed absolutely. Such a computation with digital devices, however, is complicated and time consuming relative to the more common algebraic manipulations. (See ref. 4.) Fortunately, in the present application an absolute computation is not, in general, required or desired because, once $\psi$ has been established for two data points, a and b, it is necessary only to add the incremental change in $\psi$ as the curve "steps" (with perhaps 100 increments) from point a to point b. To determine the incremental change $d\psi$ in $\psi(dx,dy)$, proceed as follows:

$$\tan \psi = \frac{y}{x}$$

$$d(\tan \psi) = d\left(\frac{y}{x}\right)$$

$$\sec^2 \psi \, d\psi = \frac{x \, dy - y \, dx}{x^2}$$

$$\frac{x^2 + y^2}{x^2} d\psi = \frac{x \, dy - y \, dx}{x^2}$$

and, therefore,

$$d\psi = \frac{x \, dy - y \, dx}{x^2 + y^2} \tag{11}$$

Thus, for each infinitesimal change in x and/or y, the resulting change $d\psi$ can be computed. Furthermore, this change can be integrated over some path to obtain the total

angle $\psi$. Independence of path, singularities, and other considerations of rigor are shown in advanced calculus texts (e.g., ref. 8).

Unfortunately, the present application involves finite changes, to be denoted by $\delta x$, $\delta y$, $\delta\psi$, and so forth, instead of infinitesimal changes. To combat the truncation errors caused by the finite integration process, a finite differential or difference is derived as follows:

$$\frac{y}{x} = \tan \psi$$

and, hence,

$$\frac{y + \delta y}{x + \delta x} = \tan(\psi + \delta\psi) = \frac{\tan \psi + \tan \delta\psi}{1 - \tan \psi \tan \delta\psi}$$

$$= \frac{\frac{y}{x} + \tan \delta\psi}{1 - \frac{y}{x} \tan \delta\psi} = \frac{y + x \tan \delta\psi}{x - y \tan \delta\psi}$$

Cross multiplying gives

$$xy + x\,\delta y - (y + \delta y)y \tan \delta\psi = xy + y\,\delta x + (x + \delta x)x \tan \delta\psi$$

and, hence,

$$\tan \delta\psi(x(x + \delta x) + y(y + \delta y)) = x\,\delta y - y\,\delta x$$

When $\delta\psi$ is small,

$$\delta\psi \approx \tan \delta\psi = \frac{x\,\delta y - y\,\delta x}{x(x + \delta x) + y(y + \delta y)} \tag{12}$$

By applying $\lim_{\substack{\delta x \to 0 \\ \delta y \to 0}} \tan \delta\psi = \delta\psi$ and $\lim_{\delta x \to 0} (x + \delta x) = x$, equation (12) reduces to

$\delta\psi = \dfrac{x\,\delta y - y\,\delta x}{x^2 + y^2}$ which, as might be expected, is the infinitesimal case.

Equation (12) can also be written as

$$\delta\psi_k \approx \frac{x_k \delta y - y_k \delta x}{x_k x_{k+1} + y_k y_{k+1}} \tag{13}$$

which shows that in the denominator of the finite case, both the present (kth) value and the next ((k + 1)th) value of the independent variables are used.

Equation (13) forms the basis of the experimental model design. Its implementation is best understood by first considering equation (11) in terms of some of the better

known digital differential analyzer (DDA) techniques (refs. 9 and 10). Accordingly, the notation of figure 9 is interpreted as follows:

dy — pulse representing a small, constant increment of $y(x)$, which is counted into (integrated by) register Y

dx — pulse representing a small, constant increment of $x$, which causes $y(x)$ to be accumulated into an internal register R (fig. 10)

dz — pulse representing a small, constant increment of function $z$ ($dz$ is emitted when R overflows)

Function $z$ may therefore be obtained by counting the $dz$ pulses – that is,

$$z = \int_a^b dz = \int_a^b y\, dx$$

This rectangular integration process is displayed in figure 11.

One DDA approach to the implementation of equation (11) is the servo technique which follows. Rearranging this equation gives

$$0 = x\, dy - y\, dx - \left(x^2 + y^2\right) d\psi \qquad (14)$$

In general, if there is a change in $x$ and/or $y$, equation (14) is no longer equal to zero but is equal to some unbalanced error $\sigma$. This error can be nulled (i.e., caused to approach zero) by the generation of proper polarity $d\psi$ increments. Equation (14) can therefore be written

$$\sigma = x\, dy - y\, dx - \left(\int 2x\, dx + \int 2y\, dy\right) d\psi \qquad (15)$$

If the basic integrator of figure 9 and a few additional symbols are used, the configuration of figure 12 is evolved as the implementation of equation (15). The additional notation shown in figure 12 is as follows:

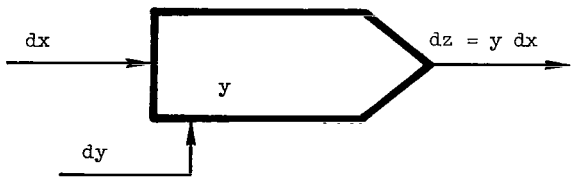



Figure 9.- Conventional DDA integrator notation.

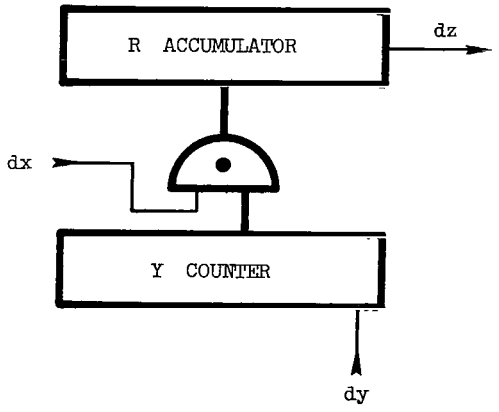



Figure 10.- Mechanization of figure 9.

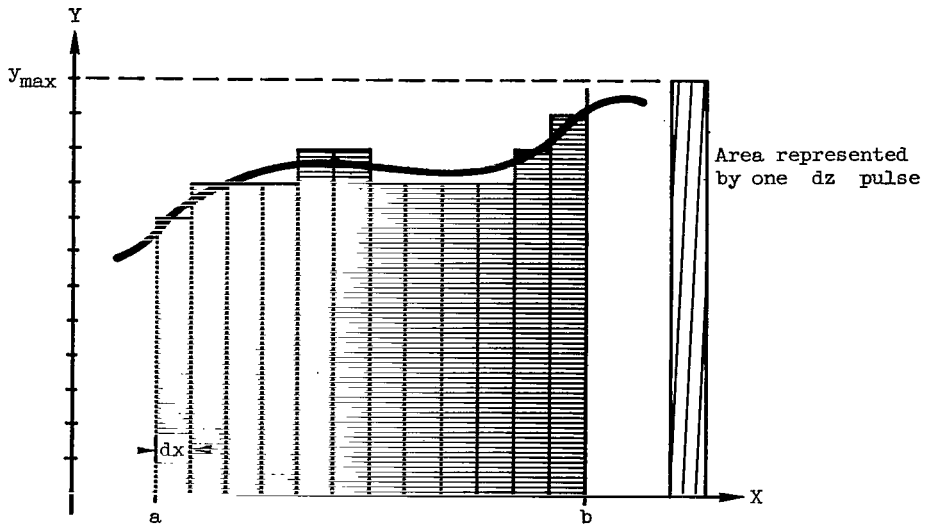



Figure 11.- Rectangular integration of y(x).

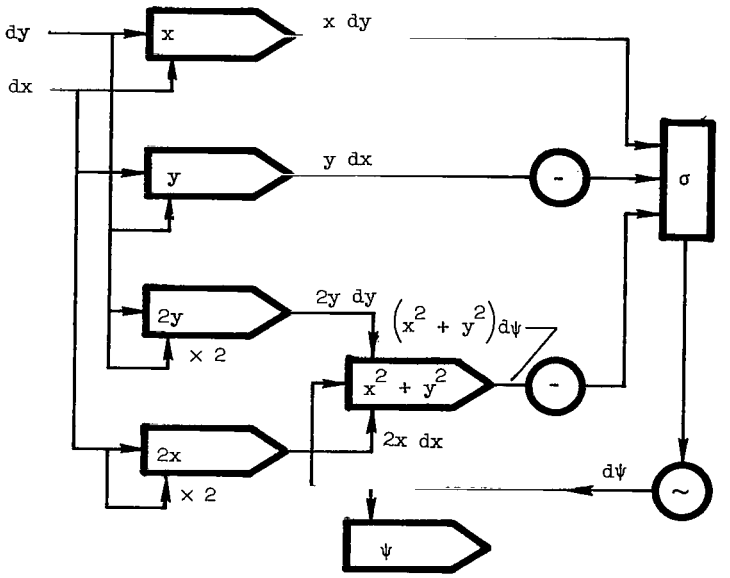



Figure 12.- Symbolic implementation of equation (15).

σ — a bidirectional counter called a summer whose input is time shared by three sources which appear as terms in equation (15)

- — indicates that y dx and $(x^2 + y^2)d\psi$ are subtracted; its function may be considered that of reversing the counting direction of σ

~ pulse generator controlled by the output of $\sigma$ (The trivalued output (+, -, 0) causes the generation of $+\psi$ increments, $-\psi$ increments, and no increments, respectively.)

×2 signifies that the input is multiplied by 2 as it enters the counter (If the counter is binary, then this multiplication is accomplished simply by connecting the input to the next most significant stage.)

The additional input to the interior integrator is time shared with its normal input.

With further extensions of the notation, equation (13) may be represented by figure 13. The only new symbol which appears is the double-headed integrator which emits, in addition to the usual kth output at one head, both kth and (k + 1)th values at the second head. Note from the figure that the top input to the integrator goes with the top output and the bottom input goes with the bottom output.

That the second head can satisfy equation (13) is seen as follows. At the end of the kth computation, assume that

$$D = x_k^2 + y_k^2 \tag{16}$$

Assume that $x$ changes 1 increment – that is, $\delta x = \pm 1$. Let this change cause, at first, only the kth value to be emitted from the second head. Then,

$$\begin{aligned} D &= x_k^2 + y_k^2 + x_k \delta x \\ &= x_k\left(x_k + \delta x\right) + y_k^2 \\ &= x_k x_{k+1} + y_k^2 \end{aligned}$$

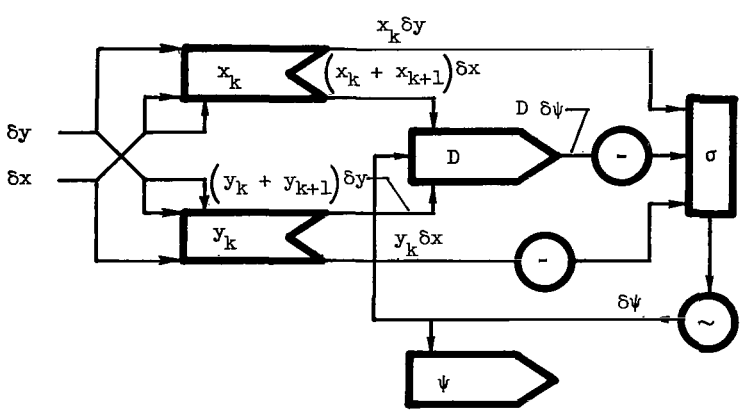



Figure 13.- Symbolic implementation of equation (13).

Since there was no change in $y$, $y_{k+1} = y_k$ and hence $y_k^2 = y_k y_{k+1}$. Therefore,

$$D = x_k x_{k+1} + y_k y_{k+1}$$

as required by equation (13). At the end of the computation, the remaining $(k + 1)$th value is emitted so that

$$D = x_k x_{k+1} + y_k y_{k+1} + x_{k+1}\delta x$$

Again, since there was no change in $y$, $y_k = y_{k+1}$ and hence $y_k y_{k+1} = y_{k+1}^2$. Therefore, at the completion of the $(k + 1)$th computation,

$$D = x_{k+1}\left(x_k + \delta x\right) + y_{k+1}^2$$

$$= x_{k+1}x_{k+1} + y_{k+1}^2$$

$$= x_{k+1}^2 + y_{k+1}^2$$

a result compatible with the assumption of equation (16). An analogous situation occurs, of course, for a change in $y$.

Figure 11 shows that the quantity represented by one $dz$ increment is relatively large. This large resolution of $dz$ is necessary to insure that no more than one $dz$ pulse is emitted for any integration step. Since the quantity represented by $dz$ is $y_{max}\,dx$, $\int dz$ can be in error by $\pm\frac{y_{max}\,dx}{2}$. To eliminate this error and to facilitate the design of the double-headed integrator, the following significant breaks with convention are incorporated in the implementation of figure 13:

(1) The integrators contain no $R$ accumulators.

(2) The integrators contain input accumulators instead of input counters; this change permits, of course, the acceptance of quantized data instead of the conventional single pulses.

(3) The summer labeled $\sigma$ in figure 12 accepts quantized information instead of the conventional single pulses.

Thus, in the conventional diagram of figure 12, all interconnecting lines "carry" single pulses, whereas in figure 13 all interconnecting lines, except those labeled $\delta x$, $\delta y$, and $\delta\psi$, "carry" quantized information.

In order to check the errors associated with equation (13) as $\psi$ is integrated over various paths, a digital computer program was written. Results obtained from use of this program indicated that, in general, the accuracy was well within the requirements of the present application. For program details, see reference 4.

## OPERATIONS OF THE EXPERIMENTAL MODEL

### General Description of Model

As indicated in the system block diagram of figure 14 and the photograph of figure 15, the input to the experimental model is paper tape, coded in the format of table I by a Flexowriter. The model's output device is a standard CalComp digital incremental recorder.

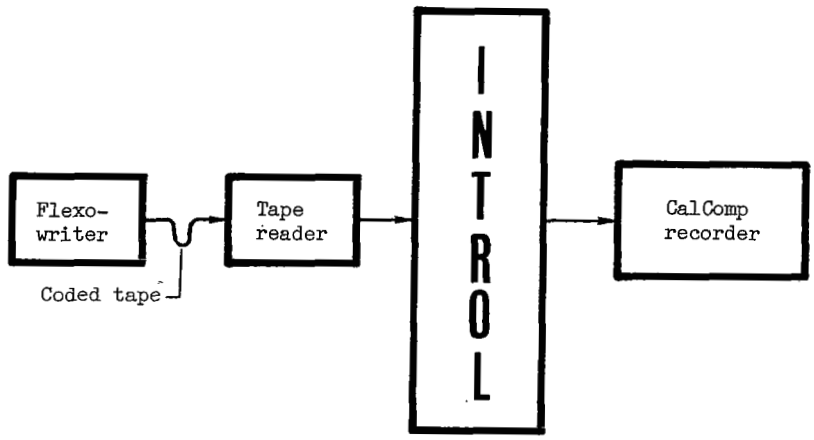



Figure 14.- Simplified diagram of the experimental model.

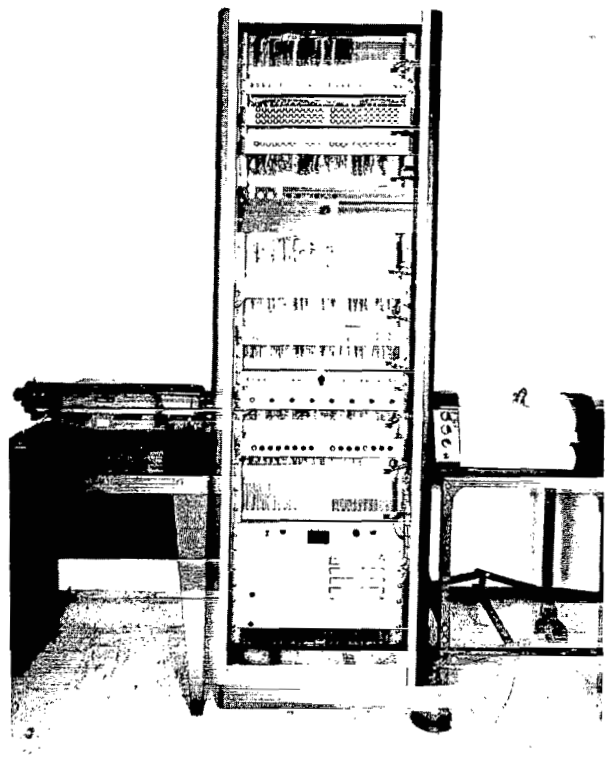


Figure 15.- Front view of the experimental model. L-65-2370

TABLE I.- TAPE FORMAT

← Tape motion

| SXXXXΔSXXXXRSXX | XXRE |
|---|---|
| x-data | y-data |

| Character | Notation | Printed symbol | Coded as X | O | C | 8 | 4 | 2 | 1 | Detection code |
|---|---|---|---|---|---|---|---|---|---|---|
| End of data | E | E | * | * | * | | * | | * | O4 |
| Carriage return | R | None | * | | * | * | * | * | | 84 |
| Space | Δ | None | | | * | | | | | $\bar{O}C\bar{1}\bar{2}\bar{4}$ |
| Sign digit | S | + | * | * | * | | | | | OC |
| | | - | * | | | | | | | $X\bar{C}$ |
| BCD digit | X | 0 | | * | | | | | | |
| | | 1 | | | | | | | * | |
| | | 2 | | | | | | * | | |
| | | 3 | | | * | | | * | * | |
| | | 4 | | | | | * | | | |
| | | 5 | | | * | | * | | * | |
| | | 6 | | | * | | * | * | | |
| | | 7 | | | | | * | * | * | |
| | | 8 | | | | * | | | | |
| | | 9 | | | * | * | | | * | |

The controller is referred to as INTROL, an abbreviation for higher order interpolation controller and, for the model, is totally contained within the relay rack in figure 15. It is implemented with the Computer Control Company 1-Mc S-PAC digital modules in which up to 28 logic cards are inserted in each of the $5\frac{1}{4}$- inch S-BLOCs seen in the photograph.

In general, the mission of INTROL is to accept input data points (similar to those of fig. 1) and direct the motion of the recorder along a smooth path (similar to the path in the fig.) through the points. The hardware which performs this operation is diagramed in figure 16.

The box labeled "input data" at the top of the figure represents the tape reader or manual data entry switches which read input data into RC. When a new data point is required, as indicated when D = 0, the point is gated by block $G_1$ into RC and the original contents of RC is shifted into RB and those of RB into RA. Switch 1 signifies that A1 can read the contents of any of the three registers as required for computing first differences necessary in the computation of arc tangents.

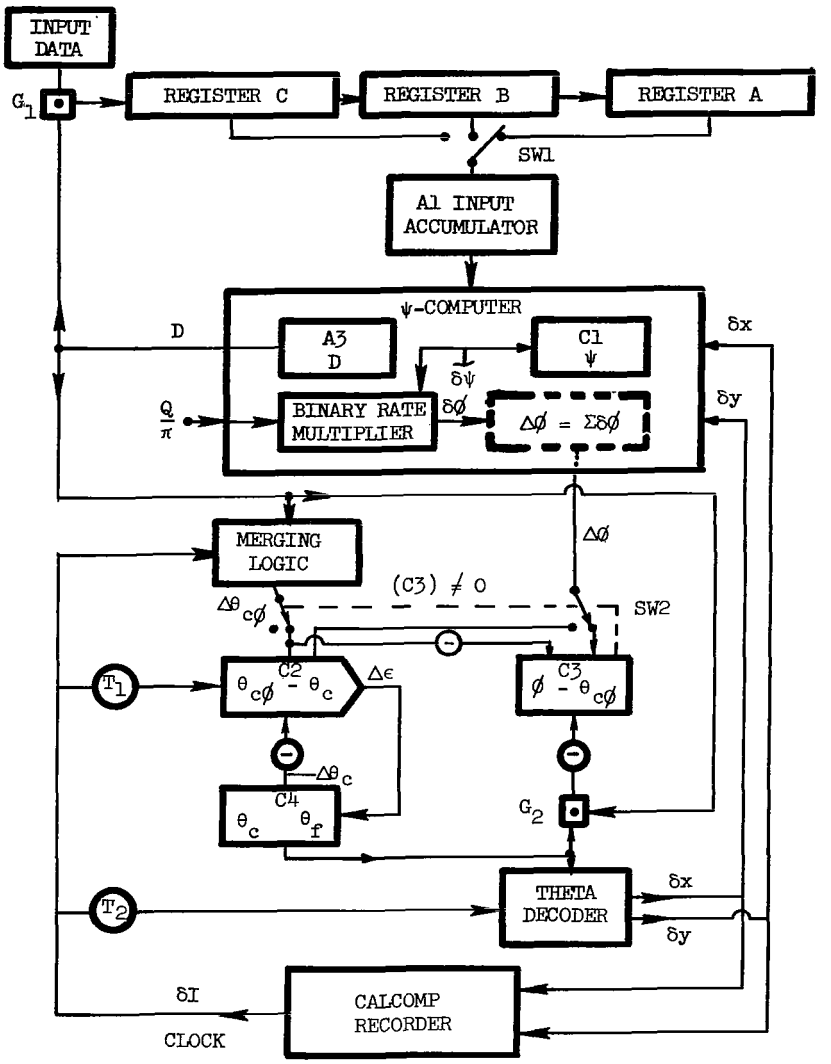



Figure 16.- Functional diagram of experimental model.

The block labeled "$\psi$-computer" represents the computer previously shown in figure 13 with the addition of a binary rate multiplier; it should be viewed now simply as a block in which $\delta x$ and $\delta y$ increments are applied and the resulting angular change $\Delta\phi$ is emitted. The D output, where D is the square of the distance to the next point, is used not only in the computation of $\Delta\phi$ but also by the merging logic to establish the instantaneous merging rate. Blocks C2, C3, and C4 are counters; C4 counts in accordance with the constraints associated with equation (2). The direction of $\theta_c$, whose resolution is $45^o$, is interpreted by the theta decoder block in terms of $\delta x$ and $\delta y$ components. These $\delta x$ and $\delta y$ increments drive the CalComp recorder and are fed back to the $\psi$-computer for determining their effect on $\phi$ and D. Block $G_2$ is used to preset (C3) to $-\theta_c$ as an initialization, where (C3) means the quantity in C3.

## Simplified Operations of Model

An operations algorithm of the experimental model is displayed in figure 17 in the form of a flow chart. Although considerable detail is omitted, the chart expands the refined algorithm given in the section "General Theoretical Considerations" to include the operations involved with the output and merging functions subsequently developed in that section.

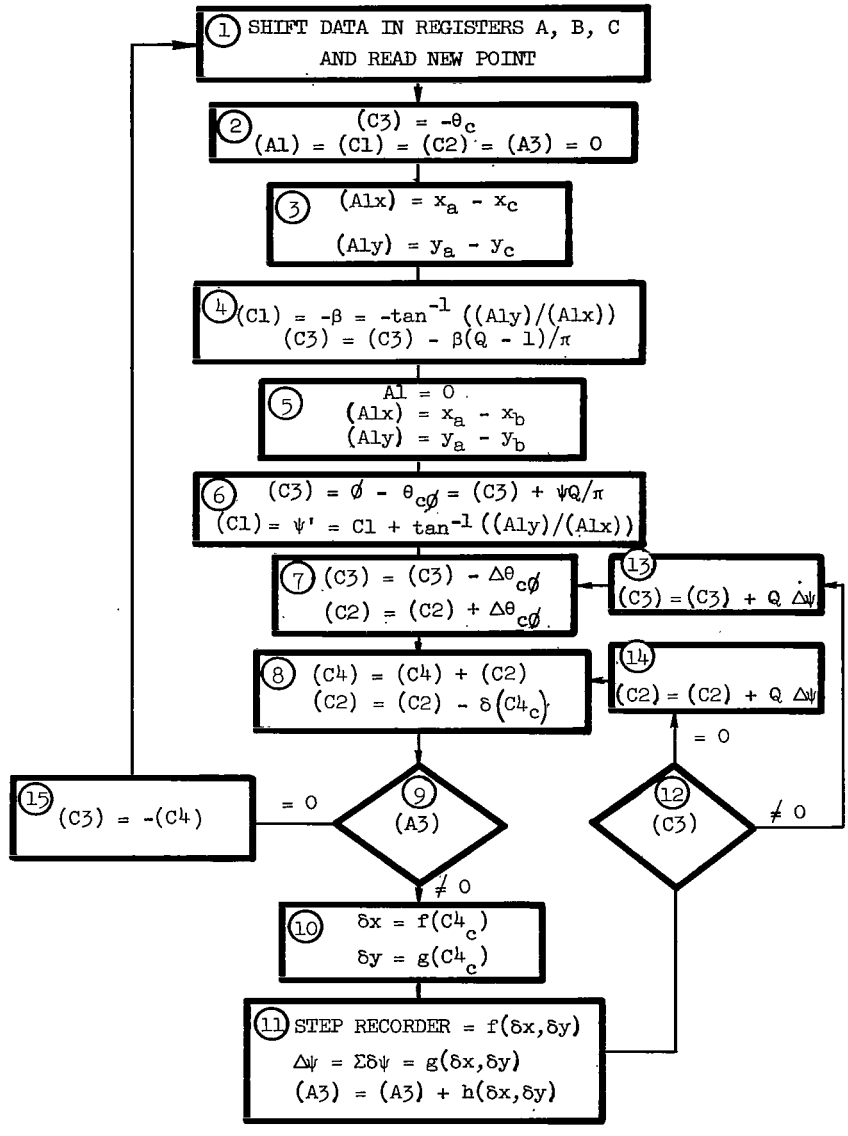



Figure 17.- Simplified operations flow chart of experimental model.

The operations algorithm for the hardware of figure 16, which is indexed with the numbered boxes of the flow chart of figure 17, is as follows (assume that the curve in fig. 1 has been generated up to point a and that signal D therefore has been reduced to zero):

① Shift data from RB to RA and from RC to RB and read a new point of the curve into RC. (The points are plane Cartesian and hence the registers and input accumulator A1 are actually implemented in x-y pairs.)

② Reset C1 and C2 and A1 and A3; preset (C3) to $-\theta_c$.

③ Switch A1 first to RA and then to RC to accumulate $(A1x) = x_a - x_c$ and $(A1y) = y_a - y_c$.

④ Integrate, with C1 which is internal to the $\psi$-computer, $\delta\psi$ over some path to obtain $-\beta = -\tan^{-1}\frac{(A1y)}{(A1x)}$. Simultaneously, multiply the same $\delta\psi$ pulses by $-\frac{(Q-1)}{\pi}$ in the binary rate multiplier and then integrate with C3. (Thus, $(C3) = -\beta(Q-1) - \theta_c$ expressed in $\pi$ radians.[3] That the binary rate multiplier gives the desired result can be seen in step ⑥.)

⑤ After resetting A1, accumulate $(A1x) = x_a - x_b$ and $(A1y) = y_a - y_b$.

⑥ Determine, with the $\psi$-computer, $\tan^{-1}\frac{(A1y)}{(A1x)}$ by integrating, without reset, the associated $\delta\psi$ pulses in C1, making $(C1) = -\beta + \psi = \psi'$. Simultaneously, multiply these $\delta\psi$ pulses by $Q/\pi$ in the binary rate multiplier and integrate the product in C3, making

$$\begin{aligned}(C3) &= Q\psi - \beta(Q-1) - \theta_c \\ &= Q(\psi - \beta) + \beta - \theta_c \\ &= Q\psi' + \beta - \theta_c \\ &= \phi' + \beta - \theta_c \\ &= \phi - \theta_c \\ &= \phi - \theta_{c\phi}\end{aligned}$$

where $\theta_{c\phi} = \theta_c$ initially (i.e., the slope associated with point a) and $\theta_{c\phi} = \phi$ after merging is complete, at which time (C3) = 0. (C1, which contains $\psi'$, as is shown in a subsequent flow chart, is used to constrain $\psi'$ to its principal range (p. 20).)

---

[3]The conversion from radians to $\pi$ radians simplifies the theta decode logic. It also keeps the multiplying factor $Q/\pi$ less than unity (for $Q < \pi$) as required by the binary rate multiplier.

⑦ For merging, reduce $|C3|$ by $\Delta\theta_{c\phi} = f\big((A3), S[C3]\big)$, where $(A3) = D$ is, again, the measure of the distance to the next point. Simultaneously, change (C2) by the amount of $\Delta\theta_{c\phi}$. (Note that $(C2) = \theta_{c\phi} - \theta_c = \epsilon$, where, as previously explained, $\theta_{c\phi}$ merges in amounts of $\Delta\theta_{c\phi}$ from $\theta_c$ to $\phi$ and hence $\epsilon$ changes from zero to $\phi - \theta_c$, the difference between the computed direction angle $\phi$ and the actual $45^o$ resolution angle of motion $\theta_c$.)

⑧ With C4, integrate the error $\epsilon$ contained in C2 as required by equation (9). If a change in the coarse portion of C4, $\delta(C4_c) = \delta\theta_c$, occurs, feed this change back into C2 so as to update its $\theta_c$.

⑨ Since, during the first traversion of the major loop, $(A3) \neq 0$, proceed.

⑩ With the theta decoder, resolve $\theta_c$ into the two orthogonal increments $\delta x$ and $\delta y$.

⑪ Using the bipolar pulses representing the $\delta x$ and $\delta y$ increments, step the CalComp recorder and increment the $\psi$-computer (so that it updates $\psi$ and D).

⑫ Since merging would not normally be complete (i.e., $(C3) \neq 0$), proceed.

⑬ Change (C3) by $\Delta\phi = Q\,\Delta\psi$, where $\Delta\psi = \Sigma\delta\psi$ for a given $\delta x$ or $\delta y$ increment, and then enter the major loop at step ⑦.

⑭ If merging is complete in step ⑫ (i.e., $(C3) = 0$, change (C2) by $Q\Delta\psi$ and then enter the major loop at step ⑧.

Eventually the test at step ⑨ will show $(A3) = 0$ at which time the entire operation from step ① is repeated.

## Detailed Operations of Model

The complete operation of the experimental model consists of the following programs:

(1) $\psi$-computer subroutine

(2) main program

(3) input program

The $\psi$-computer subroutine is called only by the main program. The input program runs simultaneously with the main program except for an interlock flip-flop designated G. The main program requests a new data point from the input program by setting G. After making the new point available, the input program resets G. After using the point, the main program again sets G, and so forth.

$\psi$-computer subroutine.- The $\psi$-computer, which now includes a binary rate multiplier, is treated by the main program as a subroutine (fig. 18). It is called with the statement:

Call $\psi$-comp ($\delta x, \delta y$); $\Delta\phi$) where $\delta x, \delta y$ increments are input arguments and $\Delta\phi = \Sigma\delta\phi$, an output argument, is the total change in $\phi$ resulting from a $\delta x$ and/or $\delta y$ increment.

The quantity $\Delta\phi$, which appears also in figure 16, is a convenience and does not actually exist – that is, there is no register which contains $\Delta\phi = \Sigma\delta\phi$. Instead, C2 and C3 actually utilize the individual $\delta\phi$ pulses. The actual representation, however, would require a transfer between the main program and the subroutine for every $\delta\phi$ pulse iteration. Fortunately, this complication is eliminated by the fictitious $\Delta\phi$ accumulator.

Although it is not explicit in the subroutine call statement, the main and subroutine programs can each sample the other's quantities, registers, and so forth.

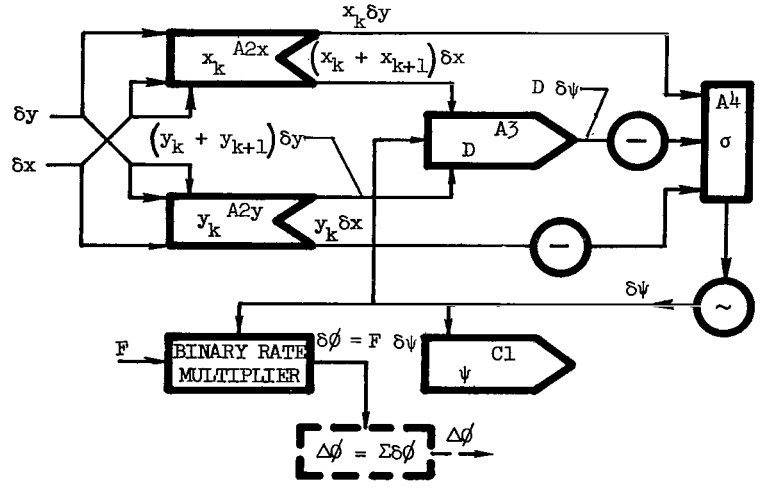



Figure 18.- Symbolic implementation of $\psi$-computer of model.

The $\psi$-computer subroutine flow chart of figure 19 should be used in conjunction with the $\psi$-computer block diagram of figure 18 which is almost the same as that of figure 13. In the following explanation of the subroutine flow chart, the numbered explanation steps are indexed with the numerals at the flow chart blocks:

① Reset the fictitious $\Delta\phi$ accumulator.

② If $|\delta y| = 1$ and $|\delta x| = 0$, proceed.

③ Reset flip-flop V. (Flip-flop V then causes computer to operate on $\delta y$.)

④ Accumulate in A3 the product of (A2y) and $\delta y$.
Accumulate in A4 the product of (A2x) and $\delta y$.
Accumulate in A2y the $\delta y$ increment.

⑤ Reset $\delta y$.

⑥ If, in step ②, $|\delta y| = 0$ and $|\delta x| = 1$, proceed with steps ⑥, ⑦, and ⑧ in an analogous manner to steps ③, ④, and ⑤.

⑨ If, in step ②, $|\delta y| = 1$ and $|\delta x| = 1$, proceed with step ⑨. If V is reset, go to step ④; if V is set, go to step ⑦.

⑩ If (A4) $\neq 0$, proceed.

⑪ Assign the sign of (A4) to $\delta\psi$.

⑫ Accumulate in A4 the negative of the product of $\delta\psi$ and (A3).

⑬ If flip-flop $\beta$ is set, proceed. ($\beta = 1$ signifies that angle $\beta$ is being computed.)

⑭ "Decrement" (C1) by $\delta\psi$ and set $F = -(Q - 1)/\pi$. (See step ⑥ on p. 32.)

⑮ If, in step ⑬, $\beta = 0$, which signifies that $\psi$ is being computed, increment (C1) by $\delta\psi$ and set $F = Q/\pi$. (See step ⑥ on p. 32.)

⑯ Accumulate in the fictitious $\Delta\phi$ accumulator the product $F\,\delta\psi$. Go to step ⑩.

⑰ If (A4) = 0, proceed with step ⑰. If V = 1, proceed.

⑱ Compute $\Delta A3 = x_{k+1}\delta x$.

⑲ Accumulate $\Delta A3$ into (A3). Go to step ②.

⑳ If, in step ⑰, V = 0, compute $\Delta A3 = y_{k+1}\delta y$. Go to step ⑲.

㉑ If $|\delta y| = 0$ and $|\delta x| = 0$, return to the main program.

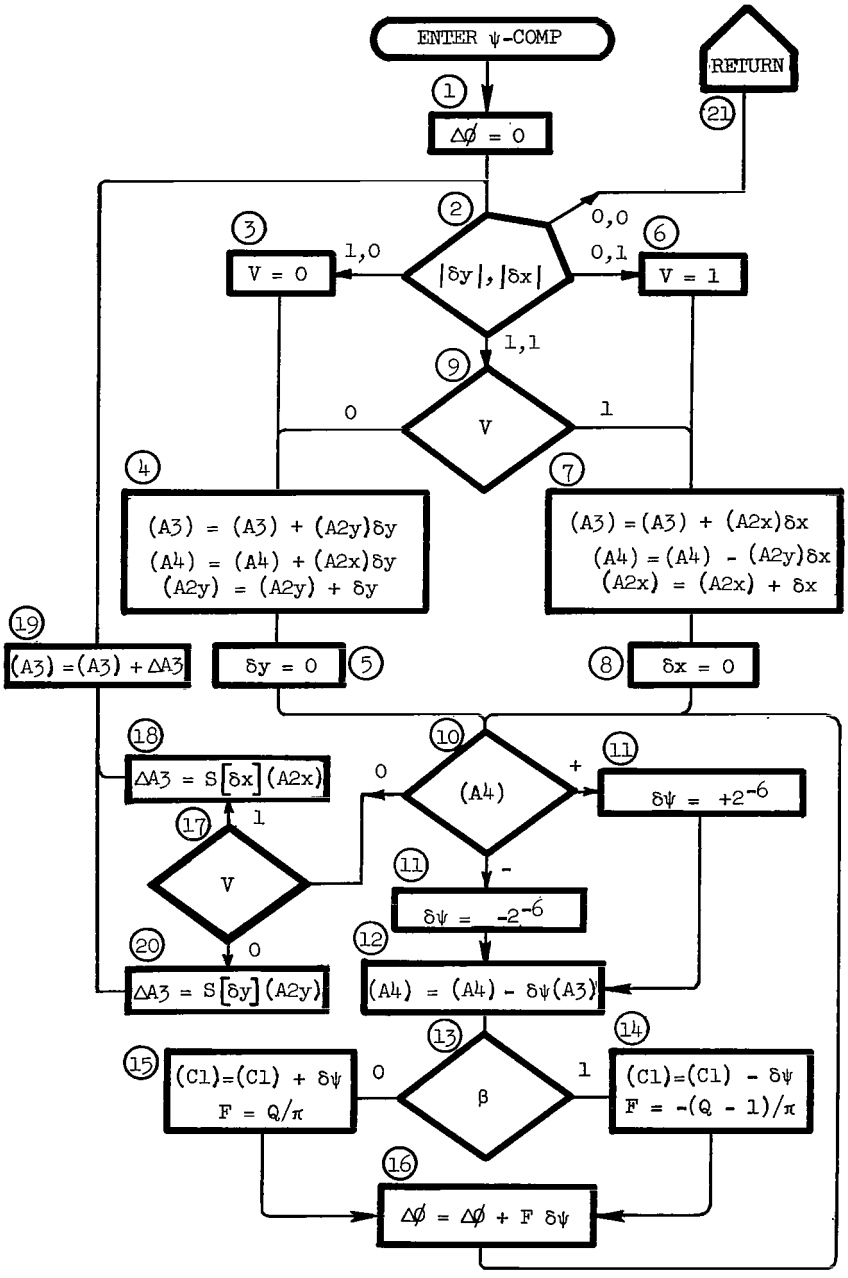



Figure 19.- $\psi$-computer subroutine flow chart.

Main program.- An appreciation of the procedure used in integrating $\psi$ is helpful in understanding the main program flow chart. Recall from equation (12) that it is important to keep the change in $\psi$ small for incremental changes in x and y. Also, the angle $\psi$ in figure 20 may be integrated by jumping from the origin to point m and then integrating to point a as was done in reference 4. This jumping avoids the singularity (ref. 8) at the origin, saves integration time, and is permissible since the path from 0 to m has no effect on $\psi$. In terms of the hardware of the experimental model, this jumping presents a problem since (A3) of the $\psi$-computer must be preset to the product $x_K^2$, a quantity which, in general, is unknown. Hence, the design of the experimental model is based on the following integration procedure:

(1) Jump exactly 12 increments in the negative x-direction.

(2) Integrate 12 increments in y in the direction of the data point (i.e., point a in fig. 20).

(3) Integrate along the x-direction toward the point until its x-coordinate is reached.

(4) Integrate along the y-direction toward the point until its y-coordinate (and hence the point) is reached.

Such a procedure avoids the singularity at the origin, permits presetting (A3) always to $12^2 = 144$, and keeps the changes in $\psi$ small.

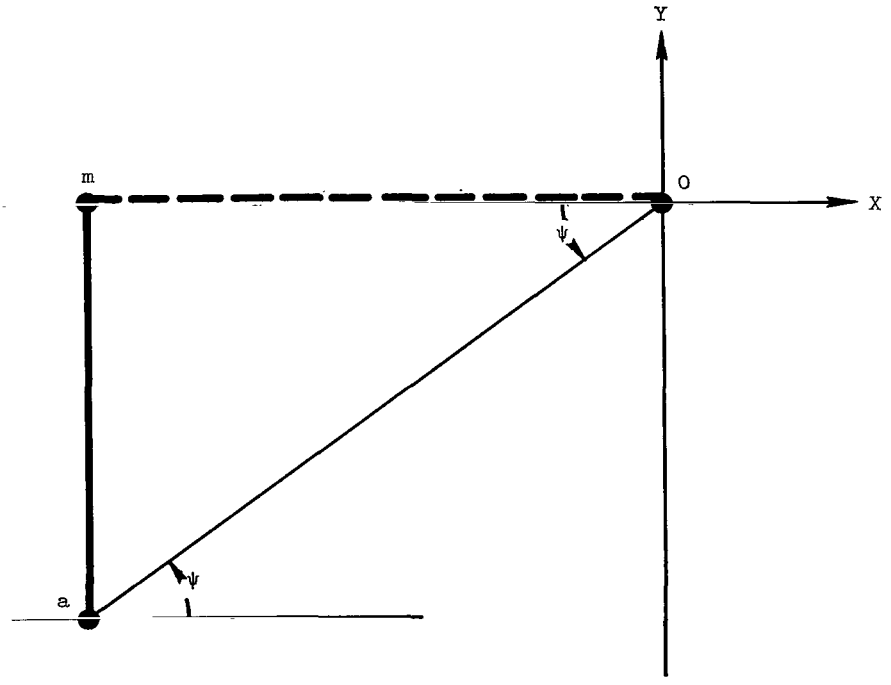



Figure 20.- Integration of angle $\psi$.

To detect when the coordinates of the point are reached as required for steps ③ and ④ in the integration procedure, (A1x) is decremented for each step in the x-direction and (A1y) is decremented for each step in the y-direction. A coordinate of the point is reached therefore when the associated accumulator reaches zero.

The main program flow chart of figure 21 should be used in conjunction with the experimental model hardware of figure 16. In the following explanation of the main program flow chart, the numbered explanation steps are indexed with the numerals appearing at the flow chart blocks:

① Select the desired type of trajectory with switch Q. Set recorder to initial position, lower pen, insert paper tape, and so forth; push start.

② Reset C3 and flip-flop L; preset U to 1 and set flip-flop G. (Flip-flop G interlocks the main and input programs which otherwise operate independently.)

③ If G = 1, proceed.

④ If end of data character E has not been read on tape, go to step ③.

⑤ If, in step ③, G = 0, which signifies RA, RB, and RC are filled, proceed with step ⑤. If U < 3, proceed.

⑥ Increment U.

⑦ Set G; go to step ③.

⑧ If, in step ⑤, U = 3, reset C1 and A1 and set flip-flop $\beta$.

⑨ Preset (A1x) to 12. (See step ① of integration procedure.) Accumulate (RAx) and -(RCx) so that $(A1x) = x_a - x_c + 12$.

⑩ Accumulate (RAy) and -(RCy) so that $(A1y) = y_a - y_c$.

⑪ Reset A2, A3, and A4.

⑫ Preset (A2x) to -12 and (A3) to $x_k^2 = 12^2 = 144$. (A2x and A3 are a part of the $\psi$-computer as shown in fig. 19. For an explanation of presetting to -12, see step ① of integration precedure.)

⑬ If $(A1y) = y_a - y_c$ is negative, proceed.

⑭ Set $\delta y = -1$ and $\delta x = 0$. Go to step ⑱.

⑮ Set J to 1. Go to step ⑰.

⑯ If, in step ⑬, (A1y) is positive, set $\delta y = +1$ and $\delta x = 0$. Go to step ⑮.

⑰ Decrement (A1y) by $\delta y$ and call $\psi$-comp $(\delta x, \delta y; \Delta\phi)$. (The $\psi$-computer begins the integration of $\beta$.)

(18) Increment (C3) by the $\psi$-computed $\Delta\phi = \Sigma\delta\phi$.

(19) Repeat steps (17) and (18) eleven more times. (See step (2) of integration procedure.)

(20) If (A1x) is negative, proceed.

(21) Set $\delta x = -1$ and $\delta y = 0$.

(22) Decrement (A1x) by $\delta x$ and call $\psi$-comp ($\delta x, \delta y$; $\Delta\phi$). Go to step (24).

(23) If, in step (20), (A1x) is zero, set $\delta x = +1$ and $\delta y = 0$. Go to step (22).

(24) Increment (C3) by $\Delta\phi$.

(25) through (29) Treat in analogous manner to steps (20) through (24). (Integration of $\beta$ is complete.)

(30) If, in step (25), (A1y) is zero, proceed with step (30). If $\beta = 1$, proceed.

(31) Reset flip-flop $\beta$, which signifies that $\psi$ is to be computed. Although redundant, reset A1.

(32) Preset (A1x) to -12. Accumulate (RAx) and -(RBx) so that (A1x) = $x_a - x_b + 12$.

(33) Accumulate (RAy) and -(RBy) so that (A1y) = $y_a - y_b$. Go to step (11).

(34) If, in step (30), flip-flop $\beta$ is reset, set flip-flop G. (C1 now contains $-\beta + \psi = \psi'$, the transformed angle.)

(35) If $|C1| > \pi$, which signifies that $\psi'$ is not in its principal range, proceed.

(36) Set J to 402. (Note that since the weight of one $\delta\psi$ pulse is $2^{-6}$ radian, 402 $\delta\psi$ pulses represent $2\pi$ radians. Therefore, 402 pulses (or integral multiples thereof) are used to bring $\psi'$ within its principal range. $\phi$ = (C3) is changed accordingly.)

(37) If (C1) is negative, proceed.

(38) Give $\delta\psi$ a negative sign.

(39) If $J > 0$, go to step (41); otherwise, go to step (35).

(40) If, in step (37), (C1) is positive, give $\delta\psi$ a positive sign. Go to step (39).

(41) If, in step (39), $J > 0$, add $\delta\psi$ to (C1) and $F\,\delta\psi$ to (C3).

(42) Decrement J. Go to step (39).

(43) If, in step (35), $|C1| \leq \pi$, proceed with step (43). If $U \geq 3$, proceed. ($\psi'$ = (C1) has been computed and confined to its principal range and $\phi$ is contained in C3.)

(44) Set the coarse portion of (C4) equal to the coarse portion of (C3). Reset $(C3_c)$, leaving (C3) = $\phi - \theta_c$. (Before the first portion of the curve is plotted, a reasonable initialization should be assigned to $(C4_c)$, the slope of the curve at the first point.)

(45) Reset C2.

(46) If (A3) > 0, proceed.

(47) Decode $(C4_c)$ into its $\delta x$ and $\delta y$ components.

(48) Increment the recorder with $\delta x$ and $\delta y$.

(49) Increment the $\psi$-computer with $\delta x$ and $\delta y$.

(50) If (C3) $\neq 0$, proceed.

(51) Increment (C3) by $\Delta\phi = \Sigma\delta\phi$ from the $\psi$-computer.

(52) Increment (C2) by $\Delta\theta_{c\phi} = f\big((A3), S[C3]\big)$, the merging quantity obtained from the $\psi$-computer. Decrement (C3) by $\Delta\theta_{c\phi}$.

(53) If (C3) changes sign, proceed.

(54) If (C3) $\neq 0$, proceed.

(55) Decrement (C2) and increment (C3) by $\pi 2^{-6}$ with a sign identical to S[C3]. Go to step (54).

(56) If, in step (54), (C3) = 0, proceed with step (56).
If, in step (53), (C3) changes sign, proceed with step (56).
Increment (C4) by (C2) and decrement (C2) by the amount of the change in $(C4_c)$. Go to step (46).

(57) If, in step (50), (C3) = 0, increment (C2) by $\Delta\phi$. Go to step (56).

(58) If, in step (46), (A3) = 0, preset (C3) = $-(C4_c)$. Go to step (3) to interpolate the next segment of the curve.

(59) If, in step (4), end of data character has been read on tape, proceed with step (59). If Z = 0, proceed.

(60) Set Z = 1. Go to step (8).

(61) If, in step (59), Z = 1, stop the operation since the complete curve is now drawn. (Note that step (60) is equivalent to reading the last point twice.)

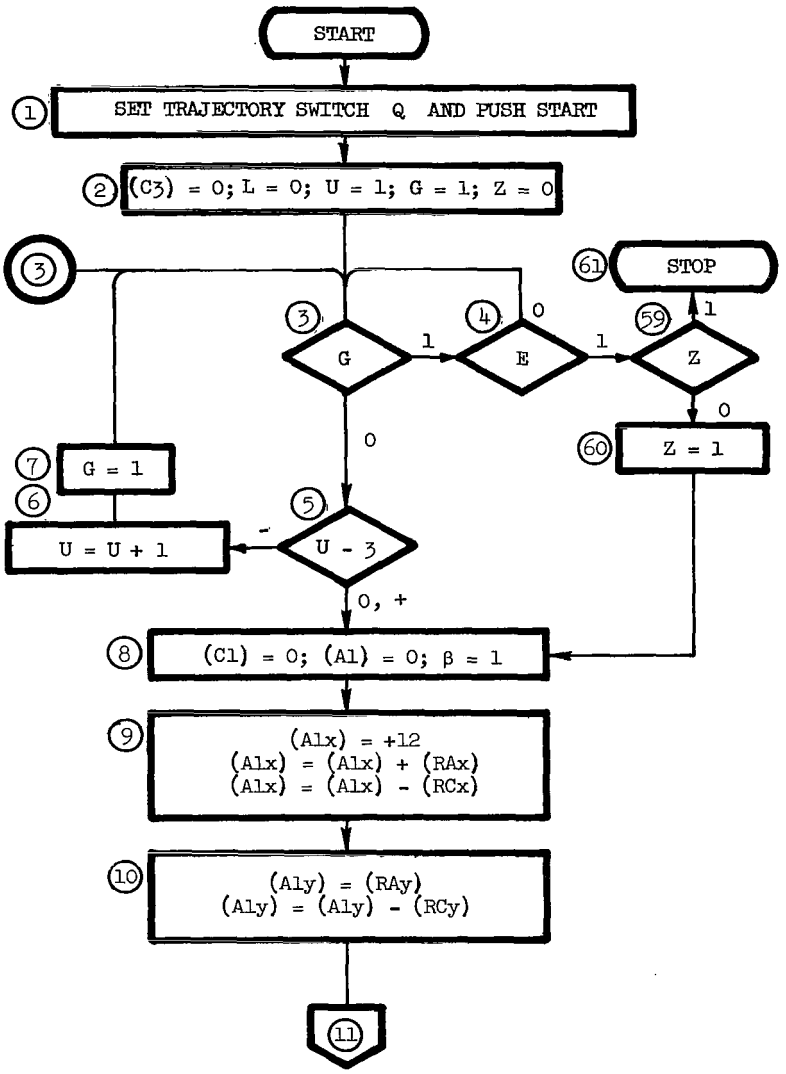



Figure 21.- Main program flow chart for model.

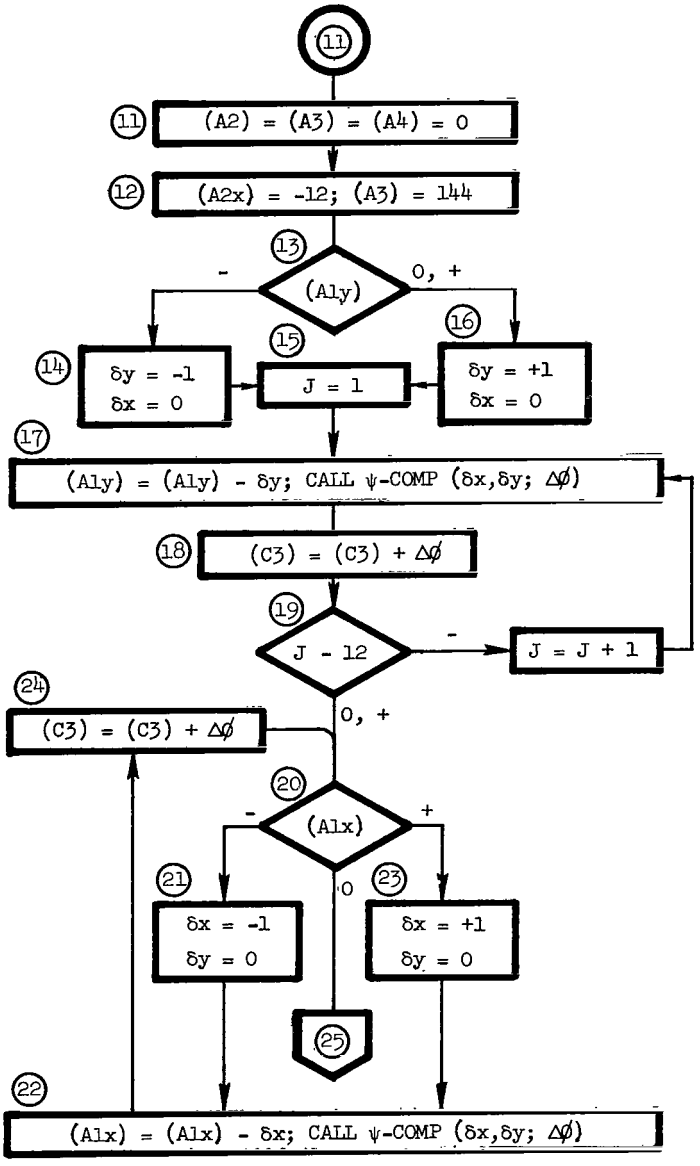



Figure 21.- Continued.

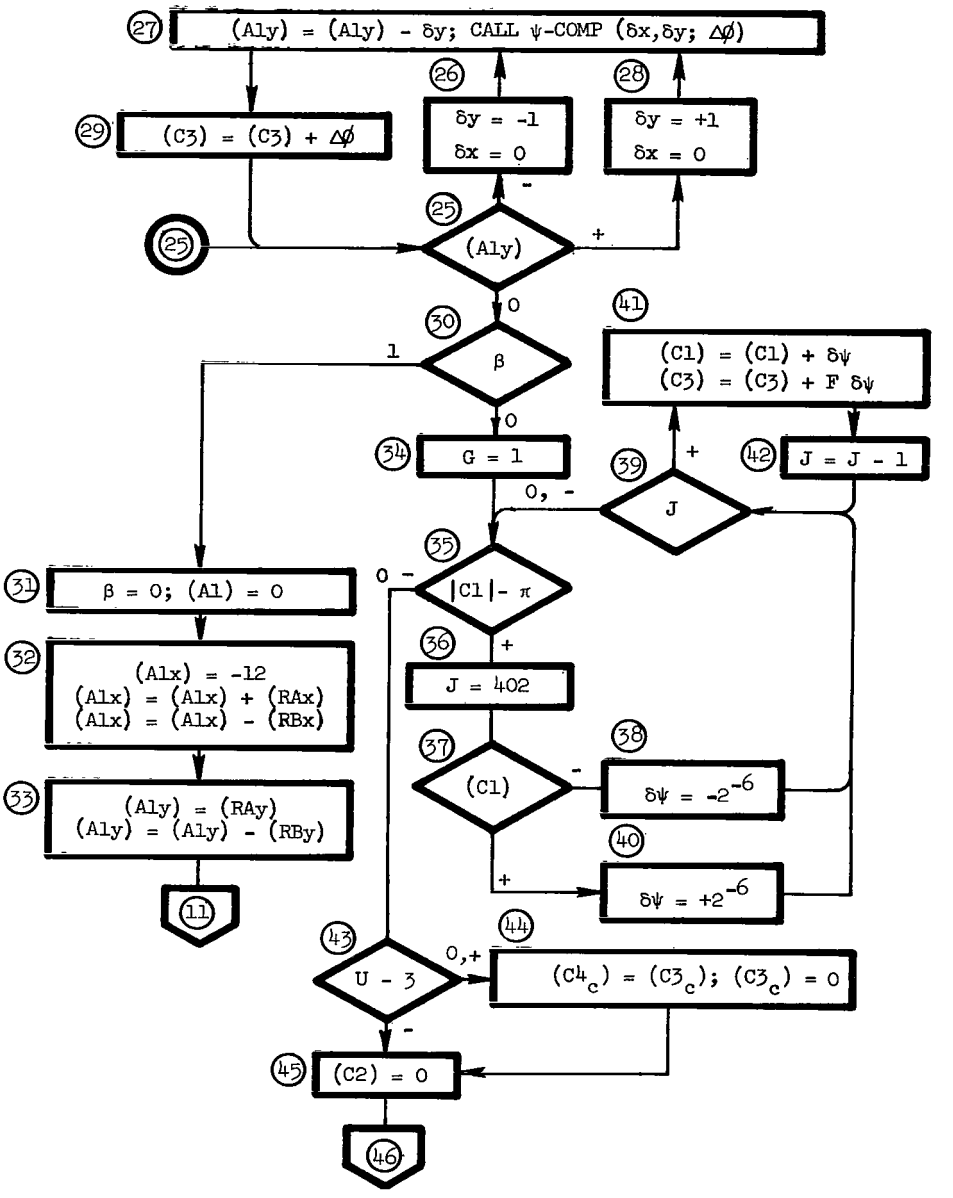



Figure 21.- Continued.

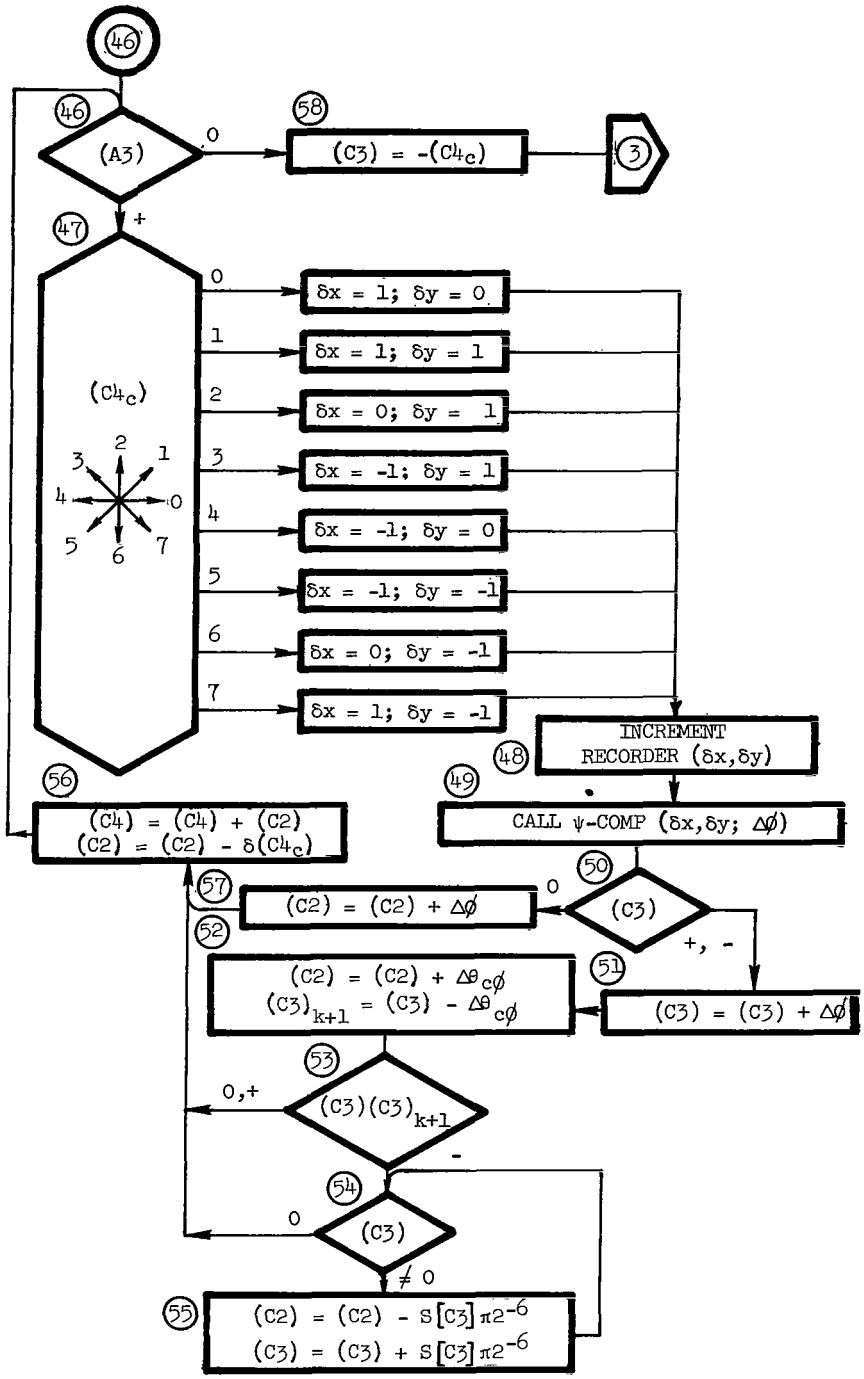



Figure 21.- Concluded.

Input program.- The input program as previously indicated operates independently of the main program except for the interlock flip-flop G which, when set by the main program, starts the input program. The flow chart of figure 22 should be used in conjunction with the input diagram of figure 23 and the tape format of table I. The following flow-chart explanation steps are indexed with the numerals appearing in the blocks on the flow chart:

① If G = 0, wait; if G = 1, which signifies that the main program has requested data, proceed.

② If model is operating in manual entry mode (i.e., not in the paper tape input mode), proceed.

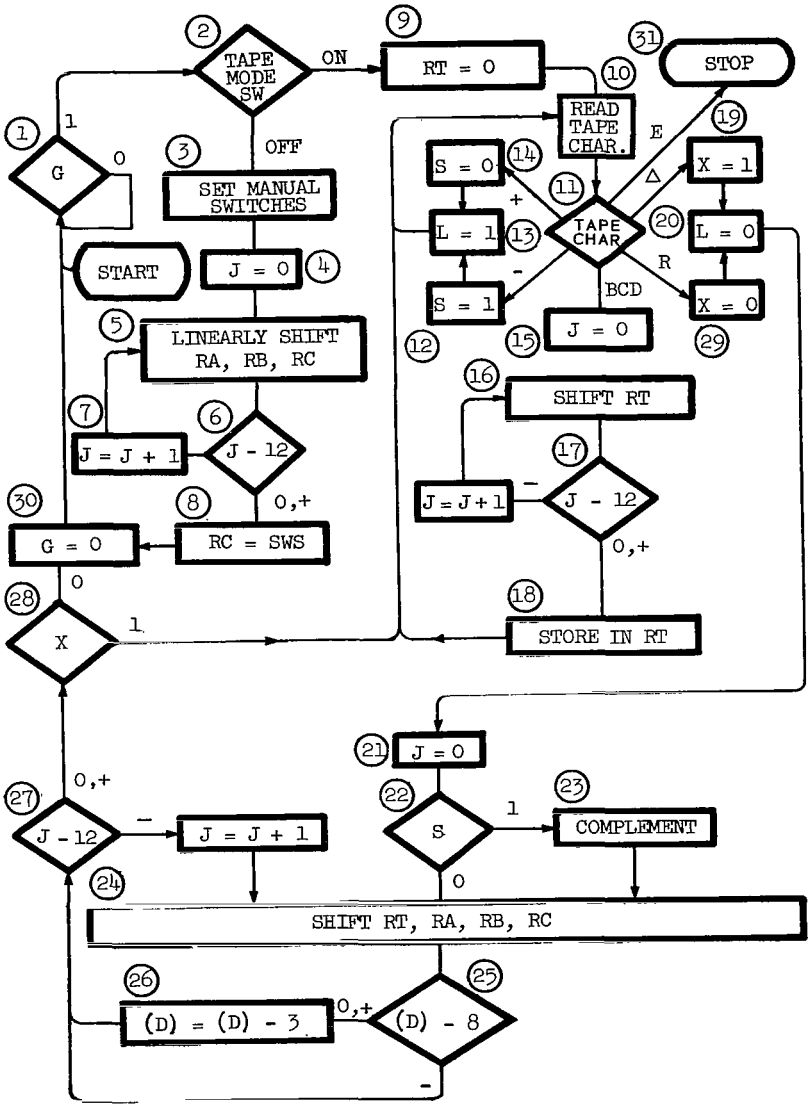



Figure 22.- Input program flow chart of model.

③ Set data point into manual switches.

④ Set J = 0.

⑤ Shift RA, RB, and RC in linear mode.

⑥ If J < 12, proceed.

⑦ Increment J.

⑧ If, in step ⑥, J ≧ 12, transfer in parallel the data entry switch settings into RC (both x and y portions).

⑨ If, in step ②, the model is operating in the tape mode, reset tape input register T and start tape reader.

⑩ Read paper tape character.

⑪ If first tape character is a negative sign, proceed.

⑫ Set flip-flop S.

⑬ Set L. Go to step ⑩.

⑭ If tape character of step ⑪ is a positive sign, reset S. Go to step ⑬.

⑮ Set J = 0.

⑯ Shift (one step) tape input register in the circular mode.

⑰ Repeat step ⑯ eleven more times.

⑱ Store BCD digit in least significant end of tape input register RT. Go to step ⑩. (Repeat steps ⑩ through ⑱ for next three characters which are also BCD digits. This procedure reverses the digits for the ensuing BIDEC operation.)

⑲ If, in step ⑪, character is Δ, set flip-flop X which connects RT to RCx.

⑳ Reset L which changes RT to linear shift mode.

㉑ Set J = 0.

㉒ If quantity is negative (i.e., S = 1), proceed.

㉓ Route quantity through two's complementing circuit.

㉔ If, in step ㉒, quantity is positive (i.e., S = 0), shift (one step) RT, RA, RB, and RC.

㉕ For any decade D containing a number greater than 7, proceed.

㉖ Decrement (D) by 3.

㉗ In step ㉕, for any decade containing a number less than 8, proceed with step ㉗. Repeat steps ㉔ through ㉖ eleven more times.

(28) If X = 1, go to step (10).

(29) If tape character of step (11) is a carriage return, reset X. Go to step (20).

(30) If, in step (28), X = 0, set G = 0, which indicates to main program that new data have been entered. Go to step (1).

(31) If, in step (11), character is E, which signifies end of tape, stop.

The main program, $\psi$-computer subroutine, and input program form the basis for the design of the experimental model.

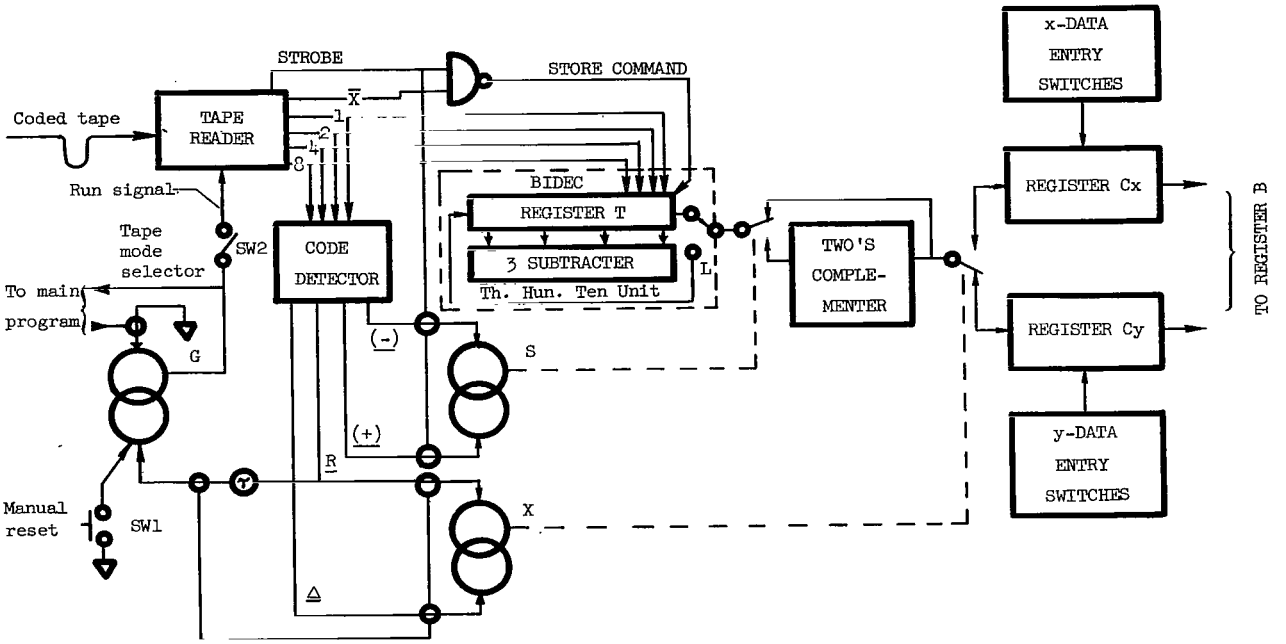



Figure 23.- Data entry diagram.

## DESIGN OF THE EXPERIMENTAL MODEL

The discussion of the design which follows points out how the subroutine, main, and input programs are implemented in the experimental model and proceeds in this same order. The logical design, in general, is presented directly in terms of NAND modules without the AND/OR design which sometimes precedes a NAND/NOR design. No attempt is made to show pin connections, duplications of circuitry, complete integration and control circuits, and so forth; instead, included is only that portion of the design philosophy which is necessary for easy extrapolation to the complete wiring design.

Figures 24 and 25 show the logic portion of the experimental model. By use of the rack layout of figure 26 the logic for the various functions performed by INTROL can be identified. In figure 24 is a temporary panel which corresponds to location S-BLOC 4 and which will be replaced by an operation flow indicator panel when INTROL becomes fully automatic.

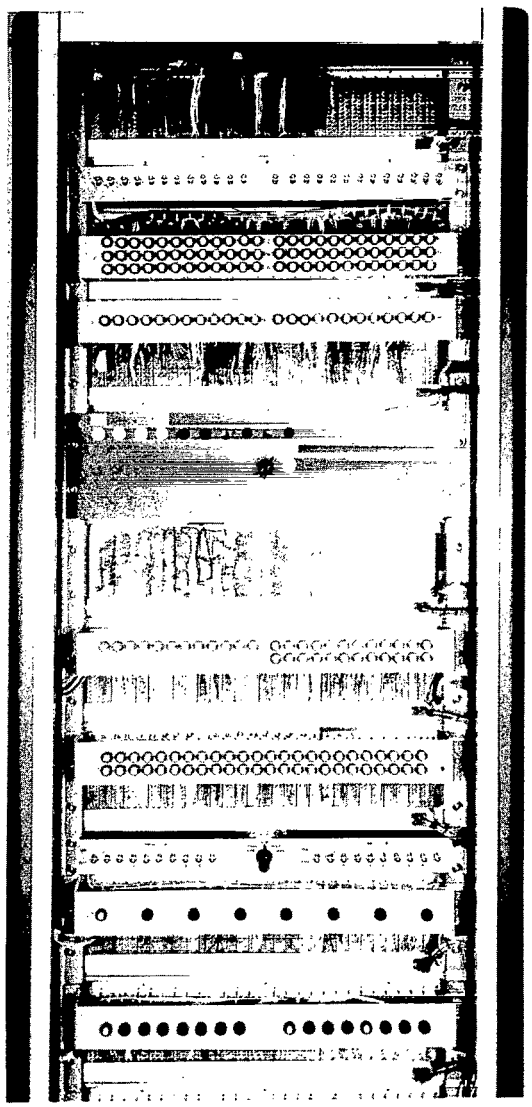


Figure 24.- Front view of INTROL logic. L-65-2367

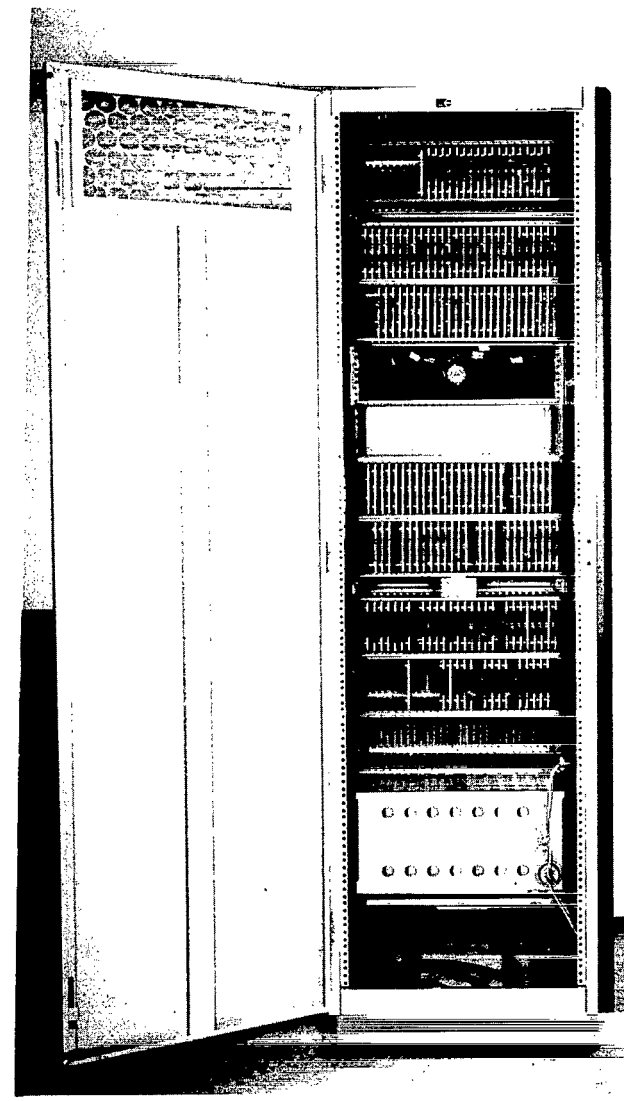


Figure 25.- Rear view of INTROL rack. L-65-2369

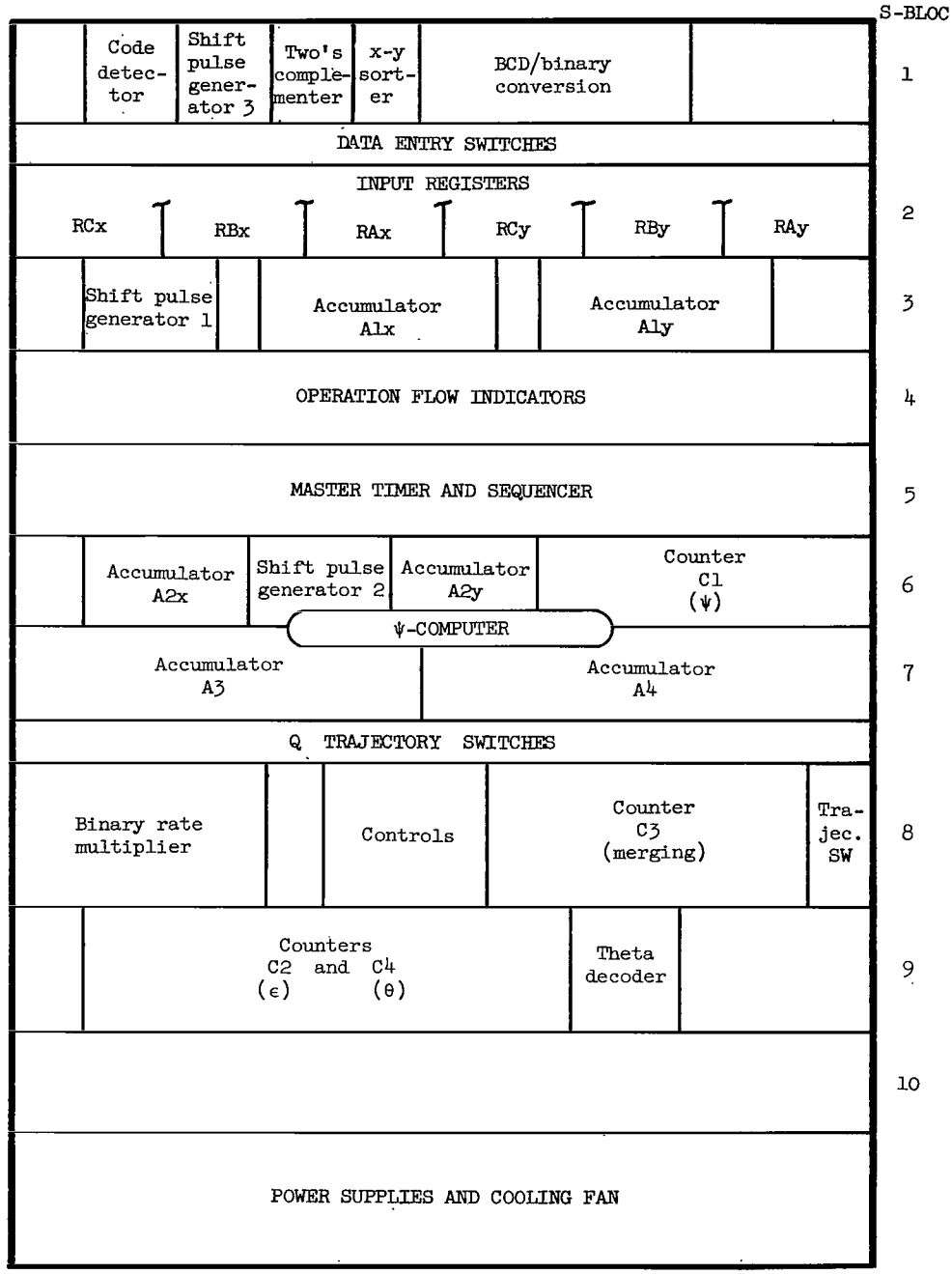



Figure 26.- INTROL rack layout.

## Design of the $\psi$-Computer

The $\psi$-computer occupies locations S-BLOCs 6 and 7, with its binary rate multiplier in S-BLOC 8. Accumulators comprise a major portion of the computer. The accumulators are similar, but they do have significant differences. Therefore, a typical accumulator is described after which the peculiarities of each are pointed out.

Typical accumulator.- In general, parallel adders are considered to be faster than serial adders. In some instances, however, the well-designed parallel adder is actually slower than the serial one that is implemented with the same speed logic. For example, the serial add time per bit for 1 megacycle logic is easily 1 $\mu$sec, whereas in a simple parallel adder 2.2 $\mu$sec was required per stage to provide for carries. Obviously unless some provision is made perhaps to eliminate this time per stage for those stages which do not actually carry, the parallel adder is slower; to make this provision, however, greatly complicates (ref. 6) the circuit. Therefore, INTROL utilizes the serial natural binary type of accumulators.

Figure 27 shows a typical serial accumulator in which the quantity B is added (or subtracted if N = 1) to (or from) quantity A and the result, replacing the old value, is stored in register A. The carry (or borrow) output $P_{k+1}$ is delayed one time cycle $\tau$ and fed back to the input.

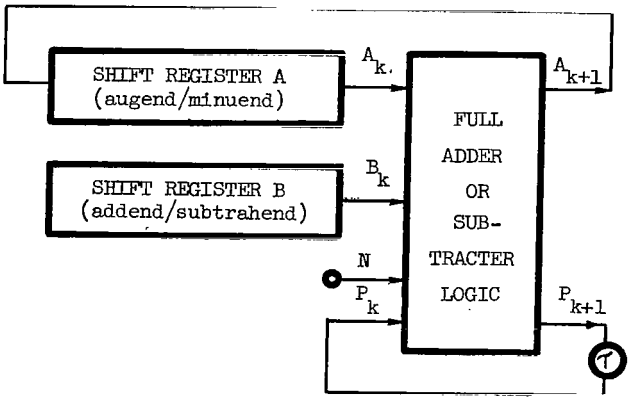



Figure 27.- Typical serial accumulator.

More typical for INTROL is the accumulator of figure 28 in which the B register is not an internal part of the accumulator and the carry (or borrow) delay is effected by the propagation flip-flop P.

Table II is a truth table for the typical accumulator. Note that the sum $A_{k+1}$ for addition where $N = 0 \equiv +$ is the same as for the difference $A_{k+1}$ for $N = 1 \equiv -$. Hence, for either addition or subtraction,

$$A_{k+1} = ABP + A\bar{B}\bar{P} + \bar{A}B\bar{P} + \bar{A}\bar{B}P \qquad (17)$$

$$A_{k+1} = A\left(BP + \bar{B}\bar{P}\right) + \bar{A}\left(B\bar{P} + \bar{B}P\right)$$

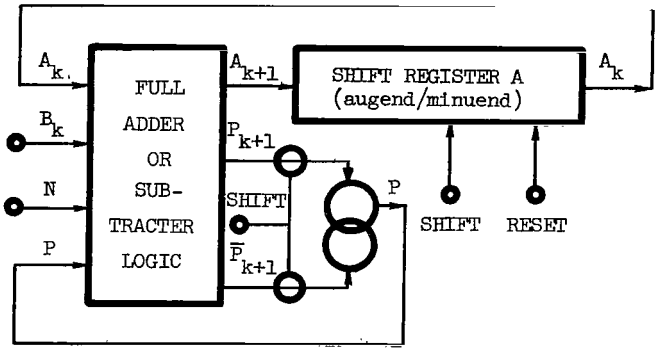



Figure 28.- Typical INTROL accumulator.

TABLE II.- TRUTH TABLE FOR TYPICAL ACCUMULATOR

| Input | | | | Output | |
|---|---|---|---|---|---|
| N | $A_k$ | $B_k$ | $P_k$ | $P_{k+1}$ | $A_{k+1}$ |
| 0 ≡ + | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 —β→ | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 —α→ | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 ≡ - | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 —α→ | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 —β→ | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 |

where the kth subscript is implicit. Let

$$H = B\bar{P} + \bar{B}P = B \oplus P \tag{18}$$

and then

$$A_{k+1} = A\bar{H} + \bar{A}H = A \oplus H = A \oplus B \oplus P \tag{19}$$

Since EXCLUSIVE-OR logic was not readily available, $A_{k+1}$ was implemented in the form of equation (17).

The truth table shows that the propagation signal undergoes only 4 transitions from $P_k$ to $P_{k+1}$ out of the possible 16. This fact leads to the use of flip-flop P and its simple logic which was obtained from figure 29 as

$$S_P = \bar{N}AB + N\bar{A}B = B(\bar{N}A + N\bar{A}) = B(N \oplus A) = BW \tag{20}$$

and

$$R_P = \bar{N}\bar{A}\bar{B} + NA\bar{B} = \bar{B}(\bar{N}\bar{A} + NA) = \bar{B}\overline{(N \oplus A)} = \bar{B}\bar{W} \tag{21}$$

where $W = N \oplus A$.

The NAND implementation of these equations is shown in figure 30.

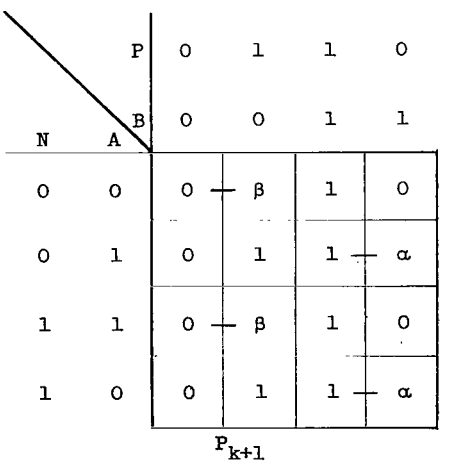



Figure 29.- Map of propagation function.

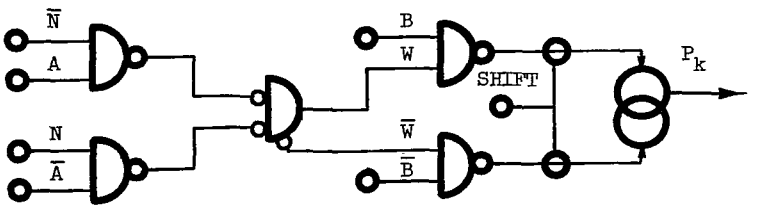



Figure 30.- Implementation of propagation function.

Accumulator 3.- The accumulator of figure 28 closely represents A3 in which shift register A is conventional (ref. 4) and comprises 24 stages. Input B is time-shared ORed with A2x and A2y outputs.

Accumulator 2.- Accumulator 2, in which the x- and y-portions are identical, is required to perform only as a bidirectional counter in the sense that its input consists only of single plus or minus pulses. However, since A2 must serve also as the B register for A3 and A4, it must be capable of shifting. Also, although not required in the present design, both $A_k$ and $A_{k+1}$ outputs are available simultaneously. Therefore, A2 is implemented as an accumulator. Its design differs from that of the typical INTROL accumulator only in that the $B_k$ input (and associated logic) is eliminated. The counting is accomplished by setting the propagation flip-flop P and then by accumulating. Therefore, for A2, where B = 0, input equations (17), (20), and (21), respectively, reduce to

$$A_{k+1} = A\bar{P} + \bar{A}P$$

$$S_P = |\delta x| \qquad \text{(for A2x)}$$

and

$$R_P = \bar{N}\bar{A} + NA$$

It is seen in figure 30 that, since $B_k = 0$, only the reset portion of the propagation function is required for A2.

The twelve stages provided in register A enable a capacity of sign and $2^{11}$ counts which are equivalent to a maximum distance of ±20.48 inches on the CalComp recorder. Since $(A3) = x_k^2 + y_k^2$, the maximum value of

$$\begin{aligned}(A3) &= \left(2^{11}\right)^2 + \left(2^{11}\right)^2 \\ &= 2^{22} + 2^{22} \\ &= 2^{23}\end{aligned}$$

Therefore, 23 natural binary stages plus a sign stage are provided. (The sign bit in this application actually serves only as a check since it should never indicate negative.)

Accumulator 4.- Accumulator 4 is a typical INTROL accumulator except that the input is time shared by three sources. For the nulling operation (eq. (14)), the nulling direction is opposite to the sign of (A4) $\left(\text{i.e., } N[A4] = \overline{S[A4]}\right)$ and shall be considered complete when (A4) changes sign. This change is detected by observing the propagation flip-flop P after each accumulation; a change is implied only when P is found in the set state.

In order to establish the sign and bit weight of A4, the following symbol definitions are used:

$L_{RJ}$ length of (i.e., number of bit positions or stages in) RJ where J is any alphabetical character

$W_{\delta u}$ weight associated with least change in u where u is any variable or (AJ)

To accommodate A3, $L_{A4} \geqq 24$. Since A4 is nulled after each input cycle, it need be no larger than A3. It is recalled that $W_{\delta A3} = 2^0$. A resolution requirement of $<1^o$ was imposed on the experimental model; hence, $W_{\delta\psi}$ was made $2^{-6} \approx 0.9^o$. Therefore, to accommodate A3, $W_{\delta A4} \leqq W_{\delta A3}W_{\delta\psi} = 2^0 \cdot 2^{-6} = 2^{-6}$. To accommodate the other two inputs, $W_{\delta A4} \leqq W_{\delta A2}W_{\delta x} = 2^0 \cdot 2^0 = 2^0$. Therefore, $L_{A4} = 24$, where six bits are located to the right of the binary point and the two inputs from A2 are scale shifted six positions as depicted in figure 31.

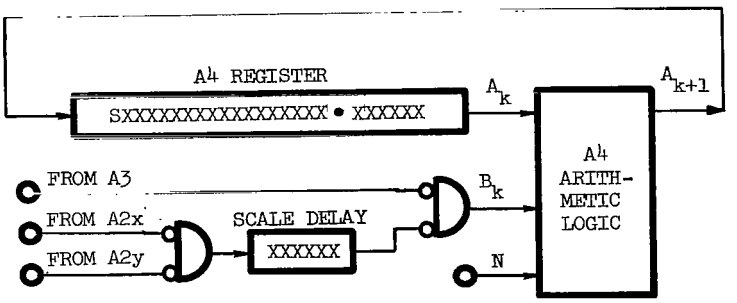



Figure 31.- $B_k$ input to A4.

Shift pulse generator.- The shift pulse generator was developed to provide the set of 24 shift pulses required for operation of the accumulators. Since the accumulators operate at speeds of 1 $\mu$sec, the generator has to count the set of pulses and make decisions at speeds approaching the upper limit of the logic used. Carry delays associated with conventional counters render such a design impractical at this speed. To minimize delays, a feedback shift register was utilized for counting. Horton (ref. 11) has shown

that a 12-state counter results if a 4-stage shift register is provided with Feedback = $\overline{A}\overline{B} + AB\overline{C}$ as shown in the boxed area at the top of figure 32. This arrangement gives the counting sequence of table III.

The circuitry at the bottom of figure 32 contains the necessary controls for starting and stopping the generation of pulses without pulse splitting. When SW1 is open, 12 pulses are generated as required for A1 in the input section. The operation of the controls is best shown in the timing chart of figure 33. The clock generates a 60-40 duty cycle pulse for best operation of the flip-flops. The flip-flops are clocked on the positive rise portion as indicated by the "action" line. For 12-pulse operation, the start pulse sets only M; $\underline{W}$ and the clock then set G at the proper portion of the clock cycle to start the generation of shift pulses.

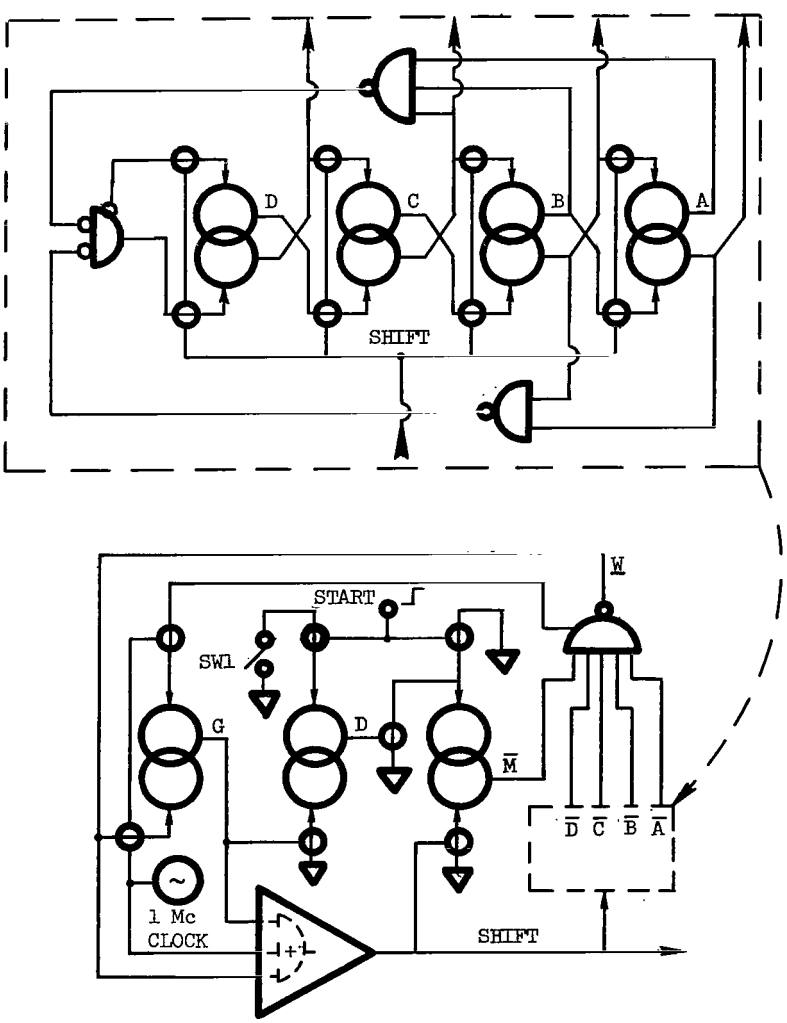
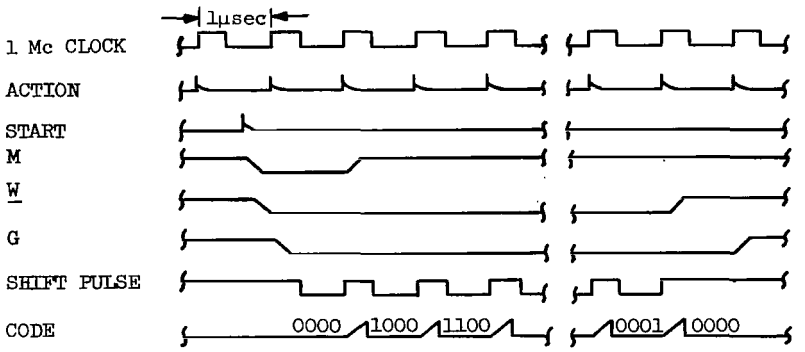


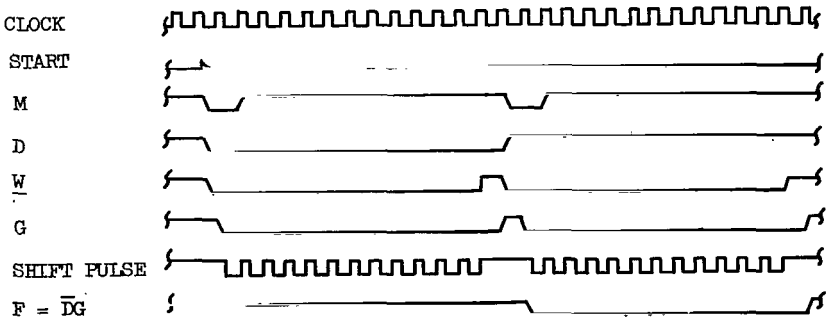

Figure 32.- Shift pulse generator.

TABLE III.- GENERATOR COUNTING SEQUENCE

| State | Stage | | | |
|---|---|---|---|---|
| | D | C | B | A |
| 1 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 0 | 0 |
| 3 | 1 | 1 | 0 | 0 |
| 4 | 1 | 1 | 1 | 0 |
| 5 | 0 | 1 | 1 | 1 |
| 6 | 0 | 0 | 1 | 1 |
| 7 | 1 | 0 | 0 | 1 |
| 8 | 0 | 1 | 0 | 0 |
| 9 | 1 | 0 | 1 | 0 |
| 10 | 0 | 1 | 0 | 1 |
| 11 | 0 | 0 | 1 | 0 |
| 12 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 |




(a) 12 shift pulses.




(b) 24 shift pulses.

Figure 33.- Shift-pulse-generator timing.

When 12 of these pulses are generated, the shift register returns to its original pattern of 0000; this turns on $\underline{W}$ which immediately stops further generation. The power amplifier has three (positive logic) OR inputs and, hence, either $\underline{W}$ or G can inhibit the generation by holding the input off.

For 24-pulse operation, SW1 is closed and, hence, the start pulse sets both D and M. When D is reset in the middle of the chain, it sets M which again effects the generation of the final 12 pulses. The gap of 2 pulses in the middle of the pulse train allows time for switching to take place in other portions of INTROL. The end of a 24-pulse operation is detected by $\beta_F$, the positive transition of the signal $F = \bar{D}G$ shown in figure 33(b).

Counter 1.- The $\delta\psi$ pulses which enter A3 are integrated by C1 so that $(C1) = \Sigma\delta\psi = \psi$. Therefore C1 is bidirectional; it is modeled after the typical counter of figure 34. It is recalled that $\beta = 1$ and $\beta = 0$ signify that the angle being computed is angle $\beta$ and angle $\psi$, respectively. Therefore, the C1 operation is given by

$$N[C1] = \beta \oplus S[\delta\psi]$$

where

$$S[\delta\psi] = S[A4]$$

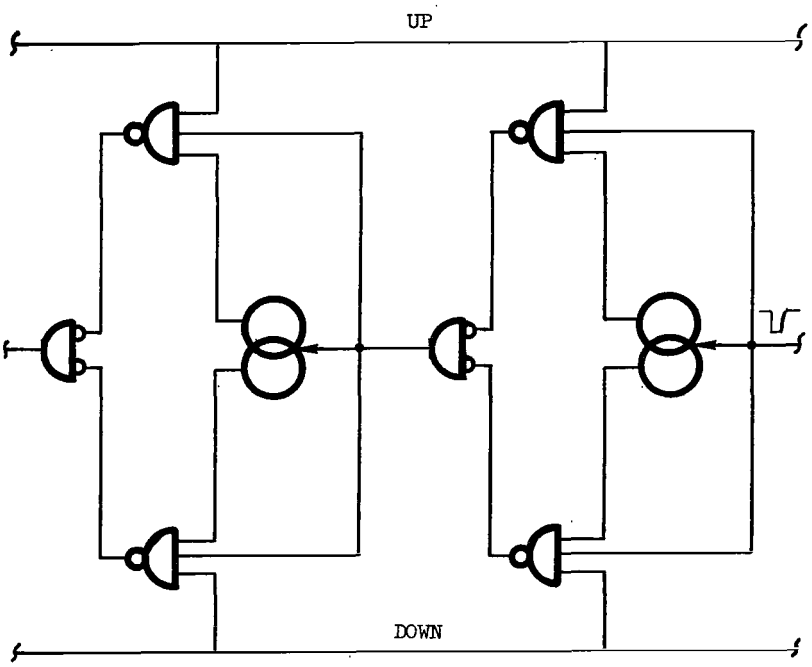



Figure 34.- Typical bidirectional counter.

Controls.- The controls required to effect the sequential operations of the $\psi$-computer subroutine are explained in the form of Boolean equations. Flip-flops X, Y, Q, and E are defined, respectively, as

$$S_X = |\delta x| \qquad R_X = M_X \beta_F$$

where $M_X$ signifies that the operation is in the X mode and F is defined in figure 33

$$S_Y = |\delta y| \qquad R_Y = M_Y \beta_F$$

$$S_Q = X\bar{Y} \qquad R_Q = \bar{X}Y$$

$$S_E = \beta_X + \beta_Y \qquad R_E = P[A4]\beta_F E$$

The E (error), X, and Y mode signals are, respectively, defined as

$$M_E = E$$

$$M_X = Q\bar{E}$$

$$M_Y = \bar{Q}\bar{E}$$

The $\delta\psi$ increments are generated by

$$\underline{|\delta\psi|} = \alpha_{\underline{E}}^{\tau} + E\overline{P[A4]}\beta_F \tag{22}$$

The shift-pulse-generator start signal is $\beta_I + |\delta\psi|$ where

$$\beta_I = \left(\bar{Y}\alpha_{\underline{X}} + \bar{X}\alpha_{\underline{Y}}\right)\bar{E} + (X + Y)\beta_E \tag{23}$$

which can be derived from the simple function $I = E + \bar{X}\bar{Y}$. This arrangement takes care of any random occurrence of $\delta x$ and $\delta y$ increments including the simultaneous occurrence. The function I is also used as

$$R_{P[A4]} = \beta_I$$

The mode signals are used to establish the proper operational signs and the proper interconnections for the four accumulators. Their logical equations are as follows:

$$N[A2x] = S[\delta x]$$

$$N[A2y] = S[\delta y]$$

$$N[A3] = S[\delta x]M_X + S[\delta y]M_Y \tag{24}$$

$$N[A4] = \overline{S[\delta x]}M_X + S[\delta y]M_Y + \overline{S[\delta\psi]}M_E \tag{25}$$

$$B[A3] = \left(\left(QA[A2x]_k + \bar{Q}A[A2y]_k\right)D[SPG2] + \left(QS[A2x] + \bar{Q}S[A2y]\right)\overline{D[SPG2]}\right)\bar{M}_E \tag{26}$$

$$B[A4] = \Big(\Big(M_X A[A2y]_k + M_Y A[A2x]_k\Big)D[SPG2] + \Big(M_X S[A2y] + M_Y S[A2x]\Big)\overline{D[SPG2]}\Big)^{6\tau} + M_E A[A3]_k \tag{27}$$

Equation (27) states that the B input of A4 is connected, during the D signal (i.e., during the first 12 shift pulses (fig. 33)), to the $A_k$ signal of A2y if the $\psi$-computer is operating in the X mode or to the $A_k$ signal of A2x if it is operating in the Y mode, and during the $\bar{D}$ signal, to the appropriate A2 sign. Before entering A4, the data are delayed by $6\tau$ or 6 shift pulses. (See scale delay register of fig. 31.) Finally, if the $\psi$-computer is operating in the error mode in which the "servo" is nulling, B[A4] is driven without delay by $A_k$ of A3.

Binary rate multiplier.- As has been indicated, the binary rate multiplier (1) performs the Q multiplication (as required, for example, in eq. (1)) and (2) changes the angular unit of measure from radians to $\pi$ radians (to simplify the theta decode logic). Since $\pi$ is an irrational number and hence cannot be precisely represented in digital form, some precision must be established for its representation in the binary rate multiplier. The resolution of $\psi$ was set at $\approx 0.9^o$ in the section pertaining to A4; this resolution represents $\approx 0.9^o$ out of $360^o$ or $\approx 1$ part in 400. In order not to deteriorate appreciably the precision limited by this resolution, $\pi$ is represented to $\approx 1$ part in 1000. Hence, the binary rate multiplier is designed to be a 10-stage device and follows the logical design of figure 35 except that it is bidirectional instead of unidirectional as shown.

It is noted in figure 35 that for I input pulses, FI pulses are emitted; $|F|$ is a binary number $<1$ – that is, $|F| = .f_1f_2f_3 \ldots f_n$, where $f_j$ are binary bits. Appropriate values of $|F|$, for various trajectories, are found in the binary columns of table IV.

TABLE IV.- INPUT CONSTANT FOR BINARY RATE MULTIPLIER

| Q | $\|F\| = Q/\pi$ | | | $\|F\| = (Q - 1)/\pi$ | | | Trajectory |
|---|---|---|---|---|---|---|---|
| | Decimal | Octal | Binary | Decimal | Octal | Binary | |
| 1.00 | .318 309 886 | .2427 | .010 100 010 111 | .000 000 000 | .0000 | .000 000 000 000 | Linear |
| 1.25 | .397 887 358 | .3135 | .011 001 011 101 | .079 579 295 | .0506 | .000 101 000 110 | |
| 1.50 | .477 464 829 | .3643 | .011 110 100 011 | .159 154 943 | .1214 | .001 010 001 100 | Cardioid |
| 1.75 | .557 042 301 | .4351 | .100 011 101 001 | .238 732 415 | .1722 | .001 111 010 010 | |
| 2.00 | .636 619 772 | .5057 | .101 000 101 111 | .318 309 886 | .2430 | .010 100 011 000 | Circular |
| 2.25 | .716 197 244 | .5565 | .101 101 110 101 | .397 887 358 | .3136 | .011 001 011 110 | |
| 2.50 | .795 774 715 | .6273 | .110 010 111 011 | .477 464 829 | .3644 | .011 110 100 100 | |
| 2.75 | .875 352 219 | .7001 | .111 000 000 001 | .557 042 301 | .4352 | .100 011 101 010 | |
| 3.00 | .954 929 658 | .7507 | .111 101 000 111 | .636 619 772 | .5060 | .101 000 110 000 | Lemniscate |
| $\pi$ | 1.000 000 000 | 1.0000 | .111 111 111 111 | .681 690 113 | .5347 | .101 111 100 111 | Maximum |

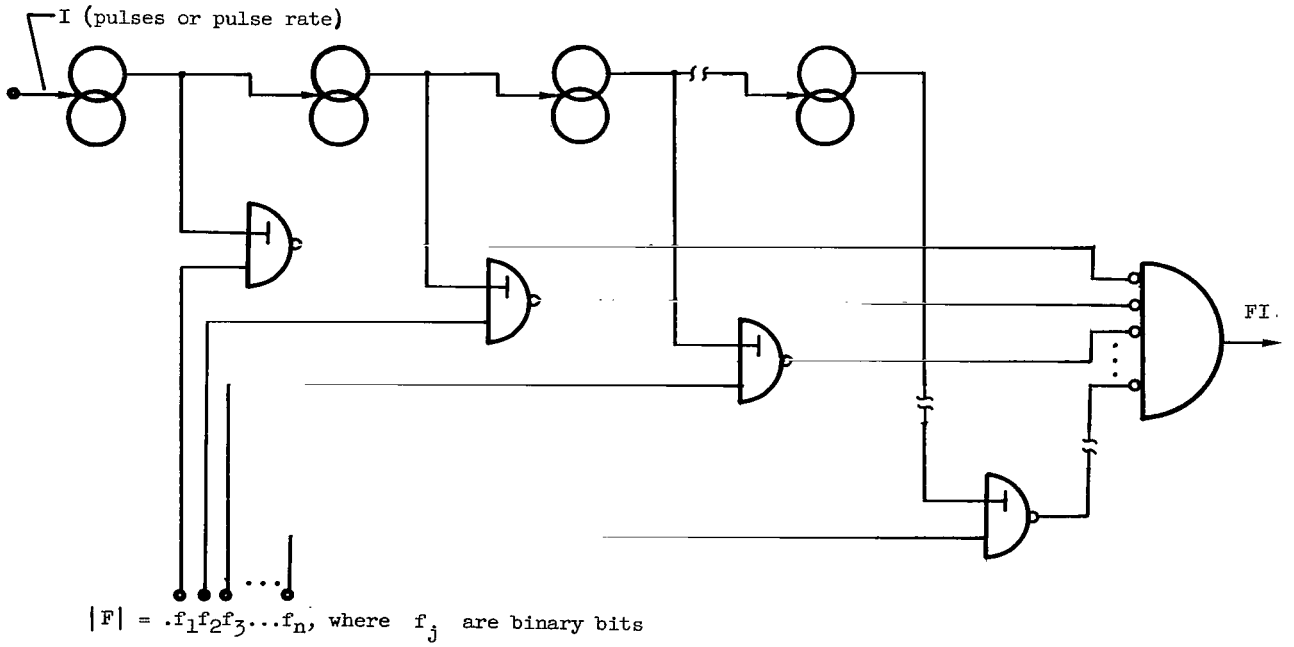



Figure 35.- Binary rate multiplier.

Counters 2, 3, and 4.- It is seen in figure 36 that $|\delta\phi|$ pulses from the binary rate multiplier go to the merging counter C3 if (C3) $\neq 0$; otherwise, the pulses go to error counter C2 which contains $\epsilon = \phi - \theta_c$. Then $\epsilon$ is integrated in C4 which contains $\theta = \theta_c + \theta_f$. As noted in the figure any change in $\theta_c$, whether from its own $S/2^{-1}$ stage or from stages $\geq 2^0$ of C2, is negatively reflected back to C2 to correct the $\theta_c$ portion of $\epsilon$.

This implementation is detailed in figure 37. It is noted that C2 differs from the typical counter of figure 34 only in that additional inputs to each stage are provided. When (C2) = $\epsilon < 0$, the value appears in two's complement notation. Therefore, since

$$\theta = \Sigma\epsilon \tag{28}$$

and (C4) = $\theta$, N[C4] = 0, always. Hence, C4 is a basic asynchronous unidirectional counter except for one change. The constraints associated with equation (28) require C4 to count or add as specified in table V. Thus the values on the left of the table are such that $\theta = \theta_c + \theta_f$, where $0 \leq [\theta_c \equiv 0(\text{mod } 1 \text{ octant})] < 8$ and $|\theta_f| \leq 0.5$ octant. The right side of the table shows the corresponding binary number. This pattern is the counting sequence of a standard natural binary counter (in which $\beta$ transitions of each stage

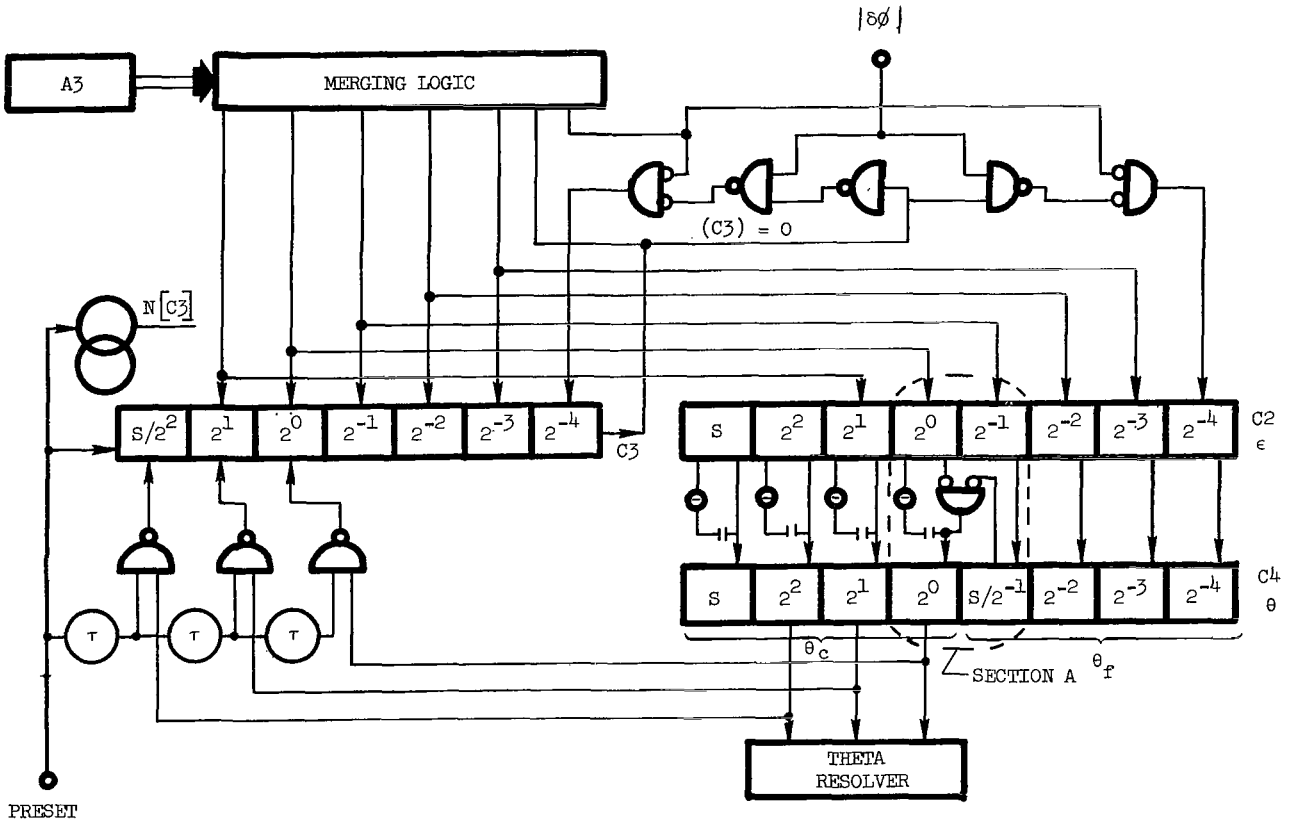



Figure 36.- Interconnections of C2, C3, and C4. (Counter dimensional unit is octant where 1 octant = 45°.)

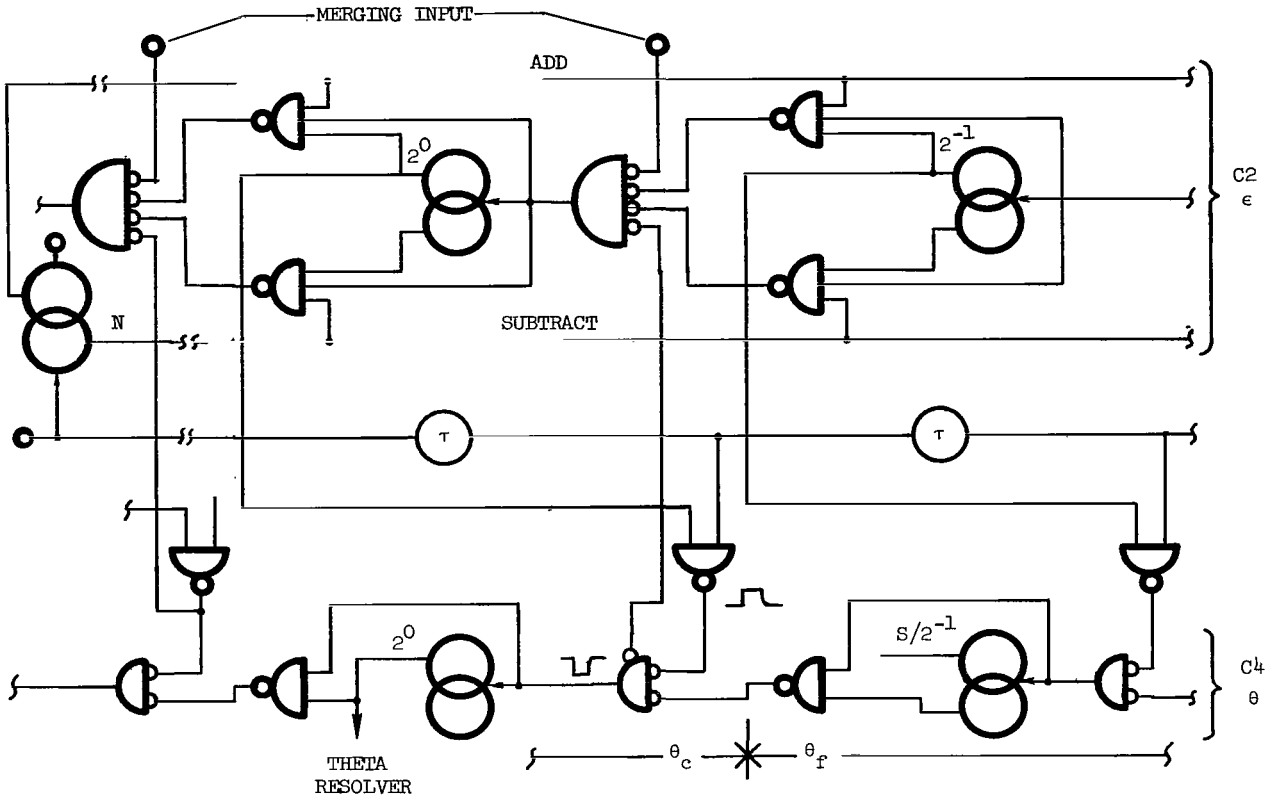



Figure 37.- Detail of section A of figure 36.

TABLE V.- COUNTING SEQUENCE FOR C4

[$\theta$, $\theta_c$, and $\theta_f$ in octants]

| $\theta$ | $\theta_c$ | $\theta_f$ | $\theta_c$ | | | $\theta_f$ | |
|---|---|---|---|---|---|---|---|
| Decimal | | | $2^2$ | $2^1$ | $2^0$ | $S/2^{-1}$ | $2^{-2}$ |
| 1.50 | 2 | -.50 | 0 | 1 | 0 | 1 | 0 |
| 1.25 | 1 | .25 | 0 | 0 | 1 | 0 | 1 |
| 1.00 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| .75 | 1 | -.25 | 0 | 0 | 1 | 1 | 1 |
| .50 | 1 | -.50 | 0 | 0 | 1 | 1 | 0 |
| .25 | 0 | .25 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7.75 | 0 | -.25 | 0 | 0 | 0 | 1 | 1 |
| 7.50 | 0 | -.50 | 0 | 0 | 0 | 1 | 0 |
| 7.25 | 7 | .25 | 1 | 1 | 1 | 0 | 1 |
| 7.00 | 7 | 0 | 1 | 1 | 1 | 0 | 0 |
| 6.75 | 7 | -.25 | 1 | 1 | 1 | 1 | 1 |

trigger the next most significant stage) except for the $2^0$ stage which triggers upon the occurrence of $\alpha$ transitions of the $2^{-1}$ stage. This simple implementation deviation can be seen in figure 37.

The design of C3 is very similar to that of C2. The addition or subtraction operations of the two counters are summarized in the following Boolean equations:

For $|\delta\phi|$ input,

$$N[C3] = N[C2] = N[C1] = \beta \oplus S[A4]$$

For $|\delta\theta_{c\phi}|$ input,

$$\overline{N[C3]} = N[C2] = S[C3]$$

For the negative feedback from C4,

$$N[C2] = 1$$

For initialization,

$$N[C3] = 1$$

## Theta Resolver

The theta resolver of figure 36 must resolve $\theta_c$ into $\delta x$ and $\delta y$ increments. Table VI is the theta resolver truth table. The X's in the table stand for "don't care" conditions. From the table it is seen that

$$\overline{|\delta x|} = \bar{A}B\bar{C} + AB\bar{C} = B\bar{C}$$

therefore

$$|\delta x| = \bar{B} + C$$

$$S[\delta x] = \bar{A}B + A\bar{B} \qquad \begin{pmatrix}\text{top} & X = 1\\ \text{bottom} & X = 0\end{pmatrix}$$

$$\overline{|\delta y|} = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} = \bar{B}\bar{C}$$

therefore

$$|\delta y| = B + C$$

$$S[\delta y] = A\bar{B} + AB = A \qquad \begin{pmatrix}\text{top} & X = 0\\ \text{bottom} & X = 1\end{pmatrix}$$

TABLE VI.- THETA RESOLVER TRUTH TABLE

| $\theta_c$, octants | | | $\lvert\delta x\rvert$ | $S[\delta x]$ | $\lvert\delta y\rvert$ | $S[\delta y]$ |
|---|---|---|---|---|---|---|
| A $2^2$ | B $2^1$ | C $2^0$ | | | | |
| 0 | 0 | 0 | 1 | 0 | 0 | X |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | X | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | X |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | X | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 |

## Merging Function

An approximation to the merging function given in figure 7 is shown in figure 38. The abscissas are related by $D = d_p^2$. The stair steps show the domains for which various $|\delta\theta_{c\phi}|$ corrections are made. The graph was used to determine the values given in table VII from which the following Boolean equations were obtained:

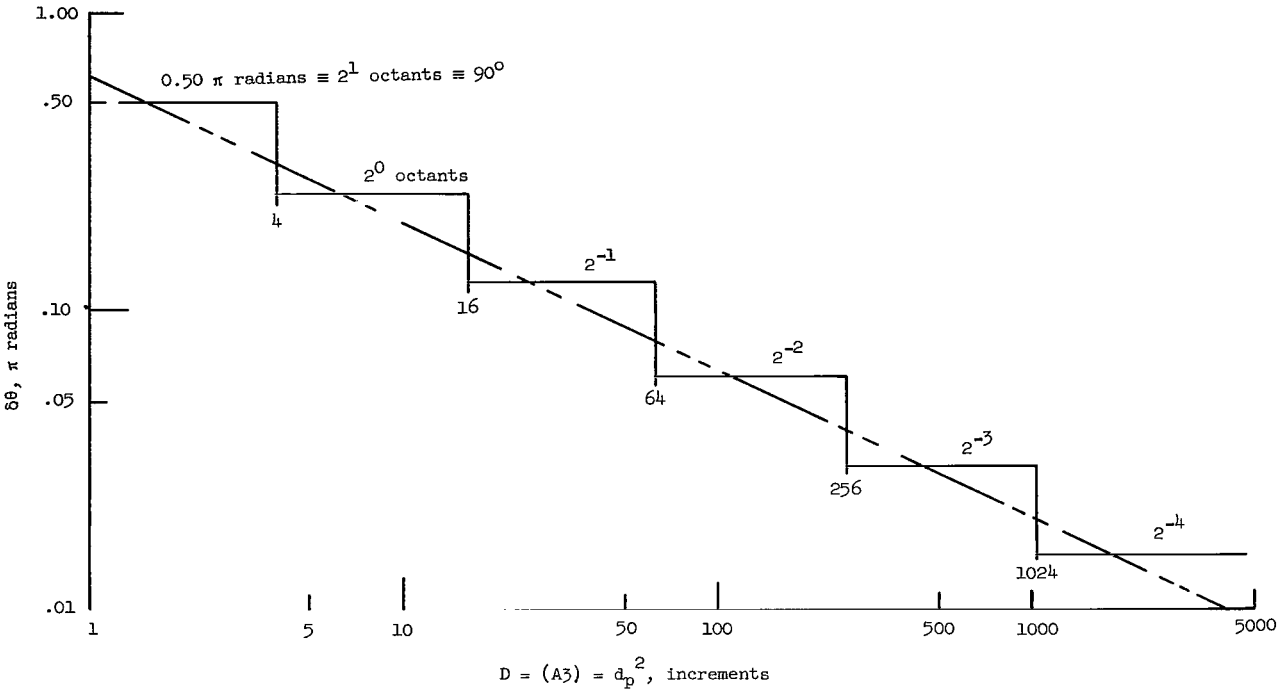



Figure 38.- Merging function approximation.

$$A = \left(|\delta\theta_{c\phi}| = 2^{-4}\right) = 10 + 11 + 12 + \ldots + 24$$

$$B = \left(|\delta\theta_{c\phi}| = 2^{-3}\right) = (8 + 9)\bar{A}$$

$$C = \left(|\delta\theta_{c\phi}| = 2^{-2}\right) = (6 + 7)\bar{A}\bar{B}$$

$$D = \left(|\delta\theta_{c\phi}| = 2^{-1}\right) = (4 + 5)\bar{A}\bar{B}\bar{C}$$

$$E = \left(|\delta\theta_{c\phi}| = 2^{0}\right) = (2 + 3)\bar{A}\bar{B}\bar{C}\bar{D}$$

$$F = \left(|\delta\theta_{c\phi}| = 2^{1}\right) = 1\bar{A}\bar{B}\bar{C}\bar{D}\bar{E}$$

Since the corrections are mutually exclusive, they are gated in parallel into the counters without carry interference.

TABLE VII.- MERGING INCREMENTS

| Most significant ON stage of A3 | $\delta\theta_{c\phi}$ correction | |
|---|---|---|
| | Degrees | Octants |
| 1 | 90 | $2^1$ |
| 2, 3 | 45 | $2^0$ |
| 4, 5 | $22\frac{1}{2}$ | $2^{-1}$ |
| 6, 7 | $11\frac{1}{4}$ | $2^{-2}$ |
| 8, 9 | $5\frac{5}{8}$ | $2^{-3}$ |
| 10 ⟶ 24 | $2\frac{3}{4}$ | $2^{-4}$ |

## Input

The input section of S-BLOCs 2 and 3 of figure 26 contains the three natural binary input registers A, B, and C and the accumulator 1 which takes the various first differences among the three input points contained in the registers.

Registers.- Since RA, RB, and RC are implemented in x-y pairs, six 12-stage standard shift registers are connected as in figure 16 so that RC shifts into RB and RB into RA. This "linear" shifting enables the registers to be updated by a new input point. The registers also operate in a "circular" mode in which the serial output feeds back into the input as required by A1.

Accumulator 1.- Accumulator 1 is represented by the typical INTROL accumulator of figure 28; it has 12 stages and uses the previously described shift pulse generator (fig. 32) which is operated with SW1 open. Since RA, RB, and RC form the various serial inputs to A1, their contents must be shifted to A1. To prevent destruction of the data during this process, they are also fed back to the registers' inputs in accordance with the circular mode.

Negative quantities in the registers are represented in two's complement form (ref. 12) and, hence, A1 operates on these quantities just as though they were positive. Therefore, the operation of A1 for RA is always addition and for the other two registers is always subtraction as required for the first differences of steps ⑨ and ⑩ in figure 21.

## Data Entry

As indicated in figure 23 two sets (12 each) of manually operated toggles are provided for entering the x and y natural binary data. After flip-flop G has been reset by SW1, a main program command pulse causes the data to be transferred in parallel into RC. Since both set and reset inputs are connected to double-throw switches, resetting the register prior to entry is not necessary. In addition to the manual switch data entry, paper tape input is provided as an alternate mode of operation. Tape, prepared on the Flexowriter in the format of table I, is interpreted by a standard tape reader. Since natural binary coding is used in the input registers, the binary coded decimal (BCD) on the tape is converted to the natural binary code before entry into RC. Also, all negative input points are converted to two's complement representation.

Code detector.- The code detector is used to recognize the non-numeric or control characters emitted by the tape reader. In INTROL these special characters serve the following functions:

Δ "space" character which on tape separates the x- and y-data and controls flip-flop X of figure 23 which channels the x and y input data to their respective C registers

(+) and (-) sign characters which control flip-flop S that "turns on" the two's complementer for negative quantities

R carriage return which stops the tape reading and sets flip-flop X for next data point

Since a seven level code is utilized and only a few of the possible $2^7$ combinations are used, many "don't care" conditions exist which permit a relatively simple synthesis of the detector. The particular detection codes used are specified in table I.

BCD to binary converter.- The BIDEC method (ref. 6) is used for the BCD to binary conversion. As related to this particular application the BIDEC algorithm is given in the section "Operations of the Experimental Model."

TABLE VIII.- TRUTH TABLE FOR "3 SUBTRACTER"

| If decade reads | | | | | Change decade to | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Binary coded decimal | | | | Equivalent decimal number | Equivalent decimal number | Binary coded decimal | | | |
| D $2^3$ | C $2^2$ | B $2^1$ | A $2^0$ | | | D $2^3$ | C $2^2$ | B $2^1$ | A $2^0$ |
| | | | | 0 to 7 | (No change) | | | | |
| 1 | 0 | 0 | 0 | 8 | 5 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 9 | 6 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 10 | 7 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 11 | 8 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 12 | 9 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 13 | 10 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 14 | 11 | X | X | X | X |
| 1 | 1 | 1 | 1 | 15 | 12 | X | X | X | X |

The specialized "3 subtracter" is synthesized by finding the logic which implements truth table VIII. Since counts 14 and 15 never occur, they are considered "don't care" conditions. From the reduction maps of figure 39, the following implementation for the four flip-flops of each decade is obtained:

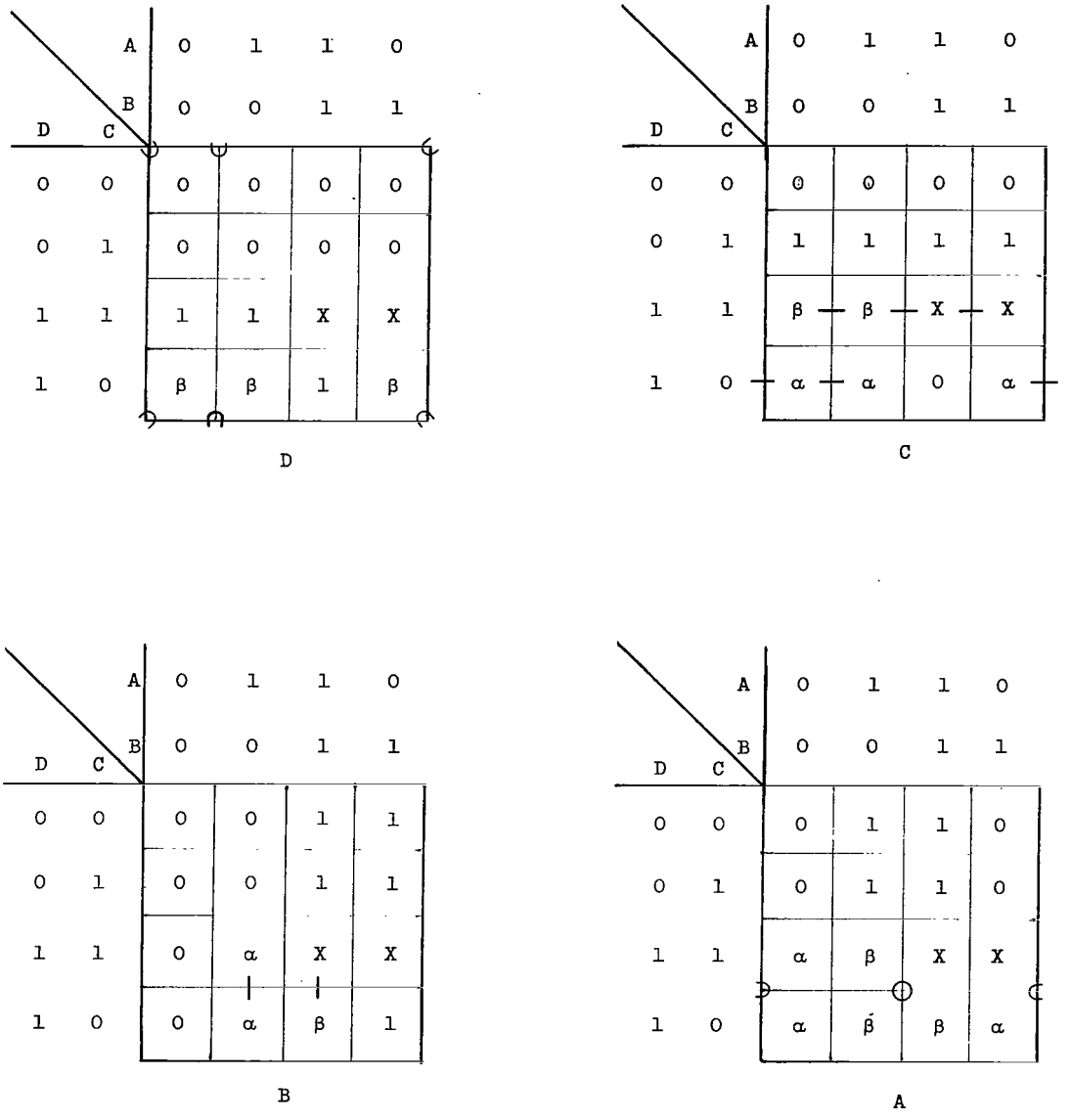



Figure 39.- Maps of BIDEC subtracter logic.

$$S_D = 0$$

$$R_D = \overline{A}\overline{C} + \overline{B}\overline{C} = \overline{C}\left(\overline{A} + \overline{B}\right)$$

$$S_C = D\overline{C}\left(\overline{A} + \overline{B}\right) = DR_D = \beta_D$$

$$R_C = CD$$

$$S_B = AD\overline{B}$$

$$R_B = ADB$$

Hence,

$$T_B = AD$$

$$S_A = \overline{A}D$$

$$R_A = AD$$

Hence,

$$T_A = D$$

It is noted that the characters are emitted from the tape reader with the most significant character leading and that BIDEC requires the characters to be shifted by RT of figure 23 with the least significant character leading. Since the shift register used is unidirectional, the required rearranging is accomplished as follows:

(1) Connect (with fictitious switch L) the serial output of the shift register to its serial input.

(2) Shift register 12 times.

(3) Read the BCD character, 4 bits in parallel, into the units position as shown in figure 23.

(4) Repeat steps (2) and (3) for a total of four characters.

Since RT is 16 bits long, its contents after reading the four characters are arranged as shown in figure 23. This arrangement is now ready for the BIDEC operation.

Two's complementer.- If the absolute value of a negative number is subtracted from zero in the typical INTROL accumulator of figure 28, the two's complement results. Thus, since the two's complementer may be considered an accumulator in which $A_k = 0$ always, equations (17), (20), and (21) reduce, respectively, to

$$A_{k+1} = B_k\overline{P}_k + \overline{B}_kP_k = B_k \oplus P_k \tag{29}$$

$$S_P = NB_k \tag{30}$$

$$R_P = \overline{N}\overline{B}_k = 0 \tag{31}$$

For the complementation of each number, P is initially reset. Equations (30) and (31) state that for complementing (i.e., N = 1) P is set the first time that $B_k$ is true in the serial operation and thereafter P remains set. Upon this hypothesis, equation (29) states that for complementing, $A_{k+1} = B_k \oplus 0 = B_k$ until the first $B_k = 1$, after which $A_{k+1} = B_k \oplus 1 = \overline{B}_k$. For positive numbers (i.e., N = 0), P never gets set and, hence, $A_{k+1} = B_k$. Therefore, the $B_k$ number does not get changed.

## EXPERIMENTAL RESULTS

In general, the experimental results are in good agreement with the theory and confirm that the mathematical technique and implementation thereof are feasible. The curves presented in this section are reproductions of those actually generated by the model and were selected because they emphasize both merits and weaknesses of the present model.

### Evaluation of Generated Curves

The coordinates of the input data points of figure 1 were determined and read into the experimental model. The generated curve of figure 40 shows close agreement with that of figure 1.

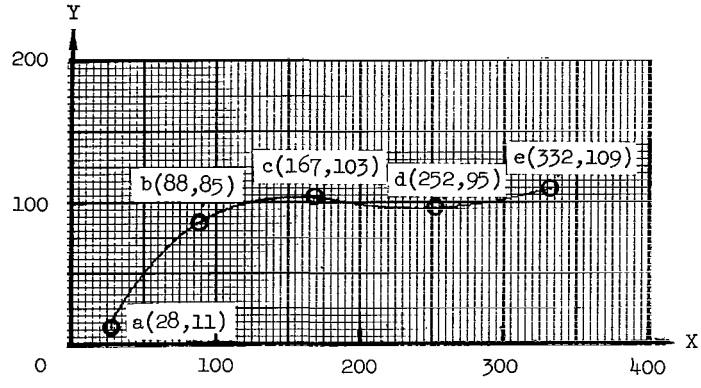



Figure 40.- Points of figure 1 interpolated by experimental model.

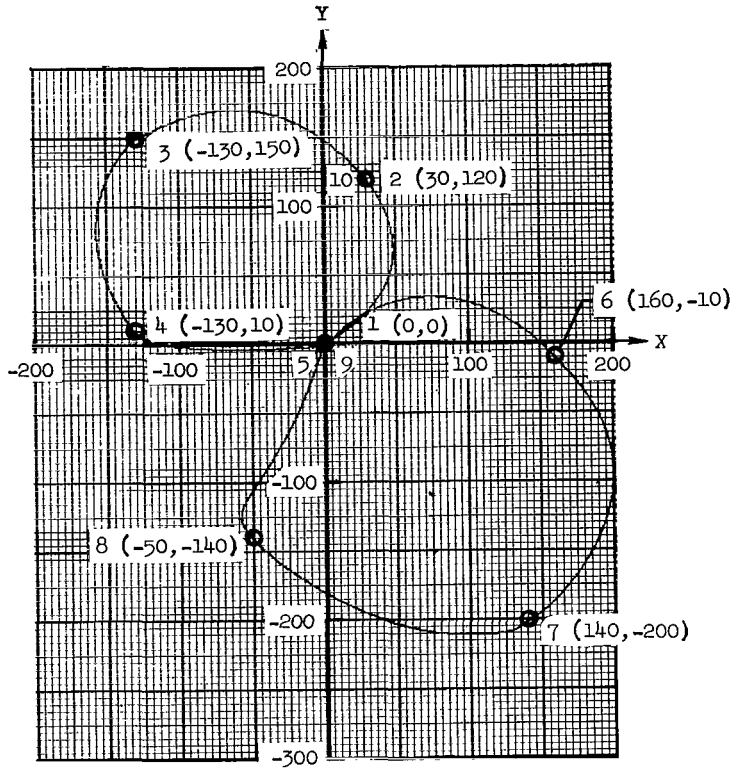



Figure 41.- The figure eight.

A considerably more difficult task is exhibited in figure 41. The figure-eight curve demonstrates the ability to

(1) draw smooth curves through widely spaced points

(2) operate on nonequal argument data

(3) produce multivalued functions

(4) process data in all four quadrants

(5) handle infinite tangents

(6) produce both positive and negative rotations

At first glance, the curve between points 8 and 9 appears to be in error. However, when the point sequence indicated in the figure is carefully considered, the distorted shape is found to be a reasonable interpolation.

The curve between points 8 and 9, however, does reveal a weakness of the experimental model. The abruptness in the turning is a result of the inability of its merging hardware to handle effectively any points which are spaced >0.7 inch. This range, however, is easily extended by adding similar stages of logic.

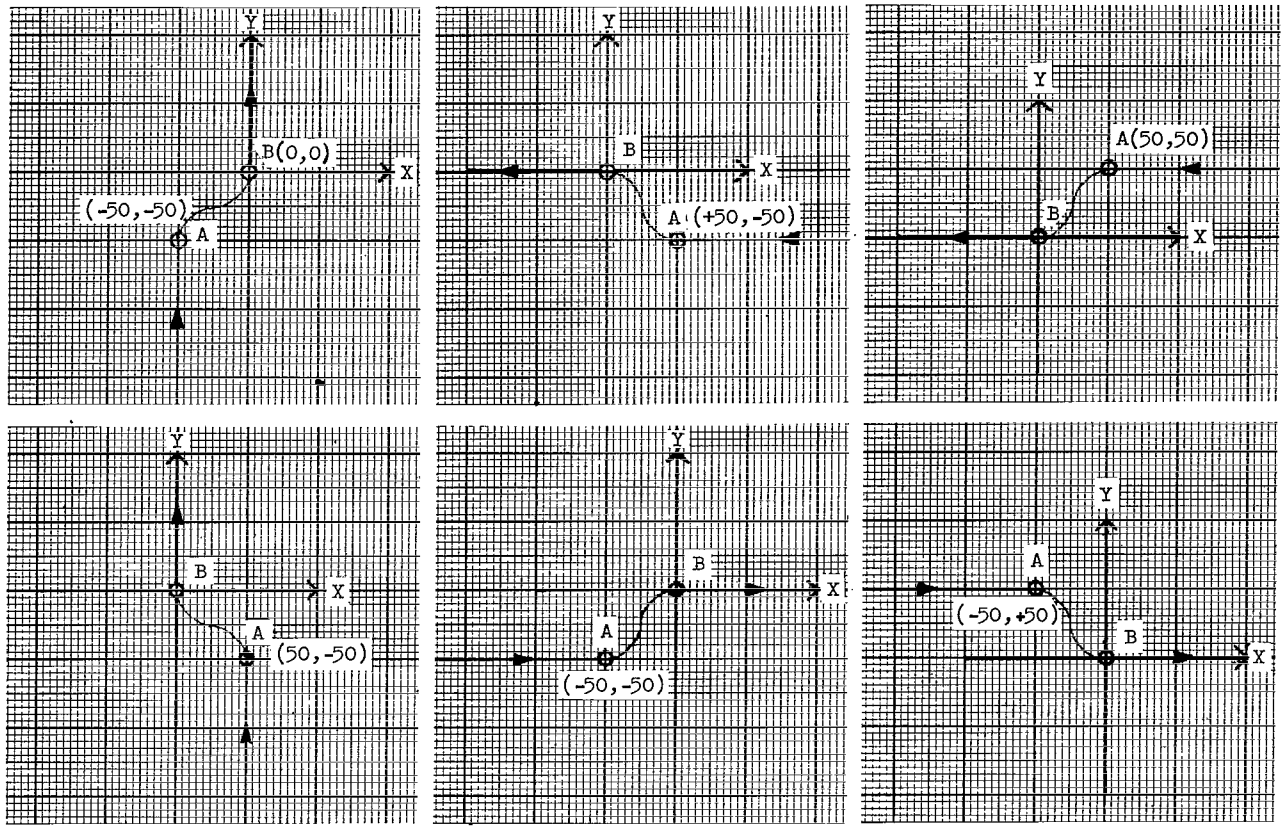



Figure 42.- Merging examples.

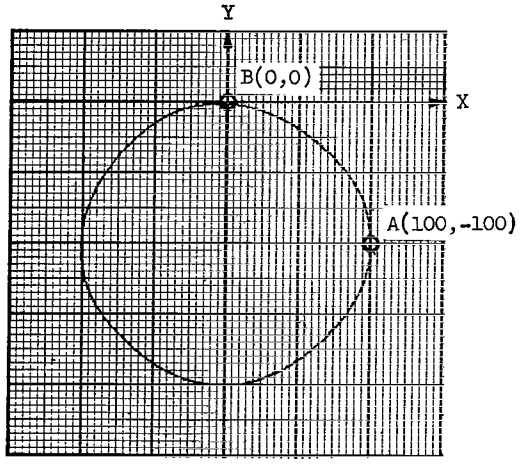



Figure 43.- Circle through two points and one slope.

Figure 42 shows some results of merging tests designed to insure satisfactory merging in all directions. The point separation of ≈0.7 inch was chosen since, as previously mentioned, this distance represents the maximum effective merging range for the model.

As predicted in the section "Derivation of Output Function," the circle of figure 43 does have four sides slightly flattened. It is recalled that this tendency to flatten results from a mathematical approximation and that for normal curve interpolation this flatness is insignificant. This particular circle was drawn clockwise, first from A to B and then from B to A. In addition to the

coordinates of A and B, the input data included the slope at B for the A to B portion and the slope at A for the B to A portion.

The curves in figures 40 to 43 were generated in the circular mode. The cardioid mode is demonstrated in figure 44. The curve was generated first from A to B over the lower path and then from A to B over the upper path. The input data consisted of the coordinates of A with respect to a coordinate system centered at B and alined with the slope (zero) of B. The upper or lower path is selected by integrating $\psi$ over a path in the upper or lower half-plane, respectively. An explanation of this selection can be seen in basic equation (1),

$$\phi' = Q\psi'$$

by observing the initial $\phi'$ for both $\psi = +\pi$ and $\psi = -\pi$. Furthermore, since $d\phi' = Q\, d\psi'$ and $Q > 0$, the sense of rotation of $\phi'$ is the same as that of $\psi'$. In general, then, as a point moves along the interpolated path, $|\psi'|$ and $|\phi'| = Q|\psi'|$ decrease monotonically, both ending with zero slope.

Careful inspection of this cardioid curve and the circular curve reveals a slight discrepancy in symmetry. This distortion results primarily from the lack of smoothing (ref. 6) in the binary rate multiplier. For applications in which it is important to correct this symmetry, as perhaps in the control of a machine tool, two or three stages of smoothing are recommended. A secondary contributor to the distortion may be the coarse resolution in several components of the system where, in the interest of simple

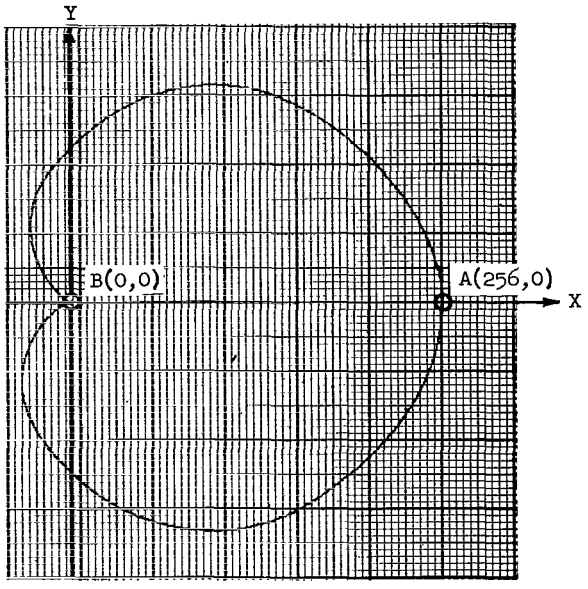



Figure 44.- Cardioid through two points and one slope.

and minimal hardware, the model was designed with fewest stages. For example, some of the computations are resolved to only $\approx 3^{o}$. In some applications, however, this resolution should be extended by a factor $\geqq 4$.

Finally, the lemniscate mode is demonstrated by the curve of figure 45. Again two points and one slope dictated the path which was drawn. It is noted that the entire path lies in the same quadrant with the starting point A. The Y-axis then represents an unstable line on which the lemniscate diverges. Thus $|\psi'|$ must be limited to $\pi/2$.

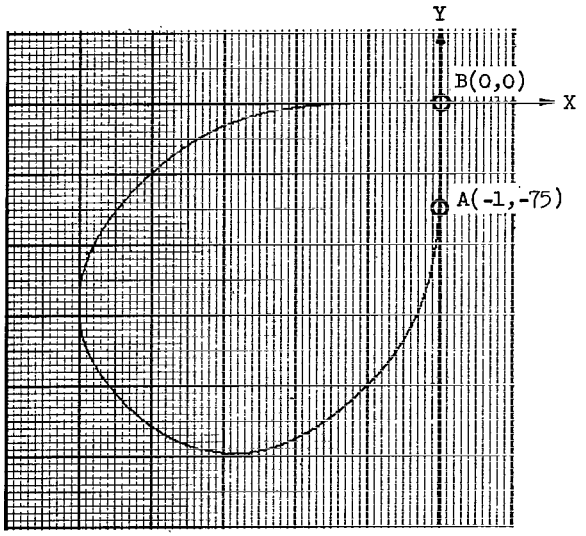



Figure 45.- Lemniscate through two points and one slope.

## $\psi'$ Boundaries

In general, the $\psi'$ limit occurs when $|\psi'|$ reaches a value such that $|\phi' - \psi'| = \pi$. This limit may be stated as a function of $Q$ as follows:

$$\pi > |\phi' - \psi'| = |Q\psi' - \psi'| = |\psi'(Q - 1)|$$

For $Q > 1$,

$$|\psi'(Q - 1)| = |\psi'|(Q - 1)$$

and, therefore,

$$|\psi'| < \frac{\pi}{(Q - 1)} \tag{32}$$

To cover all four quadrants, $\psi'$ must range over $2\pi$. This range can be effected with a maximum value of $|\psi'|$ of $\pi$. Substituting this maximum value in equation (32) gives

$$\pi < \frac{\pi}{(Q - 1)}$$

Hence,

$$(Q - 1) < 1$$

and, therefore,

$$Q < 2$$

Thus, any two points in the finite coordinate system can be connected and terminated with any prescribed slope if $1 < Q < 2$. Note that the interval is open. Therefore, for the important circular mode in which $Q = 2$, an unstable though somewhat trivial domain exists: the line for which $\psi' = \pm\pi$.

### $\psi$-Computer Error

It is recalled that the design of the $\psi$-computer is based on the approximation $\delta\psi = \tan \delta\psi$. Near the origin, however, $\delta\psi$ can be as great as $45^{\circ}$ per increment; this $45^{\circ}$ angle would be computed as $\approx 57^{\circ}$. This discrepancy of $12^{\circ}$ is prohibitive. In the model, it was reduced to an error of $< 1^{\circ}$ simply by presetting accumulator A3 to $0.01_2$ before the computation of each angle.

### Initialization of (C3)

In step ② of the operations algorithm (p. 32), each new segment $l$ of interpolation is initialized to $-\theta_c$, the angle at which the last step entered a data point. In general, this angle is not $\beta_{l-1}$, the "average" slope of the $(l - 1)$th segment into the point. Since the initialization should be $\beta_{l-1}$ and not $\theta_c$, prohibitive angular discontinuities result. Therefore, for the curves in figures 40 and 41 each initialization was manually adjusted to $\beta_{l-1}$ when $|\theta_c - \beta_{l-1}|$ was large.

The correct initialization may be implemented by using C1 which contains, at the end of a segment, a measure of the desired angle $\beta_{l-1}$. Unfortunately the unit of (C1) is radians and, hence, the quantity must be counted into the binary rate multiplier for conversion to $\pi$ radians and then recounted into C3. Step ② of the operations algorithm is corrected by the following procedure:

① Reset C2 and C3.

② Set $(C2) = -(C4_c)$.

③ Count-down (C1) through the binary rate multiplier into both C2 and C3 and apply appropriate multiplying factor and operational signs to leave $(C2) = \beta_{l-1} - \theta_c$ and $(C3) = -\beta_{l-1}$.

④ Integrate $\phi$ and simultaneously count the angle into C3 to leave $(C3) = \phi - \beta_{l-1}$.

⑤ Begin the interpolated segment.

## RESEARCH BYPRODUCTS

During the development of the subject controller, several innovations or research byproducts ensued. These byproducts are essentially separate entities which should find application in other areas, devices, or systems. The most important byproducts are as follows:

$\psi$-computer: A special purpose incremental digital computer for integrating the arc tangent angle as a function of various finite $\delta x$ and $\delta y$ input increments
The described design can be easily modified to obtain virtually any accuracy and capacity. If the $\psi$-computer is utilized as a finite difference device, its speed is great.

Bidirectional binary rate multiplier: A device which eliminates the "hysteresis" inherent in conventional binary rate multipliers when utilized in bidirectional applications

Adder with round off: A digital accumulator which outputs rounded-off information to one precision but retains a finer precision for subsequent calculations
Although developed in a natural binary system, the scheme is applicable to other weighted codes provided only that the weight of the least significant bit of the rounded portion is twice that of the adjacent lower order bit which, on turning on, carries into it. For example, in a BCD code of weight 1-2-4-5, the least significant bit (bit 1) of one decade has twice the weight as that of the most significant bit (bit 5) of the adjacent lower order decade; hence such a code is suitable.

High speed shift pulse generator: A device which, on command, generates a fixed number of fully formed pulses at a rate equal to the nominal speed of the logic with which it is implemented
Although developed with the fixed number as 24, the scheme is adaptable to virtually any quantity.

Serial two's complementer: A simple digital device for serially converting a sign-magnitude negative number to its two's complement representation (and vice versa)

## NEED FOR ADDITIONAL RESEARCH

The experimental results indicate that the merging-function—constant-Q concept developed in this research works well. Nevertheless, the concept was an expediency which does not fit harmoniously into the system. Therefore, the scheme which follows seeks to satisfy this need.

From the fundamental relation $\phi' = Q\psi'$ (eq. (1)),

$$\phi - \beta = Q(\psi - \beta)$$

At the beginning of a segment,

$$\phi = \alpha$$

and

$$\psi = \psi(A,B)$$

Therefore,

$$Q = Q(A,B,\alpha,\beta)$$

that is, Q may be considered a function of the four independent variables A, B, $\alpha$, and $\beta$. This characteristic is demonstrated in figure 46 where, for the specific points A and B, a Q-family of curves is drawn for various values of $\alpha$. If the coordinate system of this figure is considered to be the transformed system, then the $\beta = 0$ slope is generalized to any slope by the transformation itself. Thus, for any two points A and B with slopes $\alpha$ and $\beta$, respectively, a value of Q is found for use in basic equation (1) to enable generation of a smooth segment. The catch is that the smooth curve is not always the desired one. In general, the desired curve results if the direction of $\alpha$ is such that $|\psi'|$ decreases. This condition is satisfied in substantially all segments usually encountered. For an S-shape segment, however, the condition fails and the generated segment is not the desired connection.

One approach is to ignore the failure condition since it seldom occurs. Another approach is to restrict the input data spacing to prevent the failure condition. A third and best approach is to find a simply implementable solution which copes effectively with the failure condition and gives the desired path.

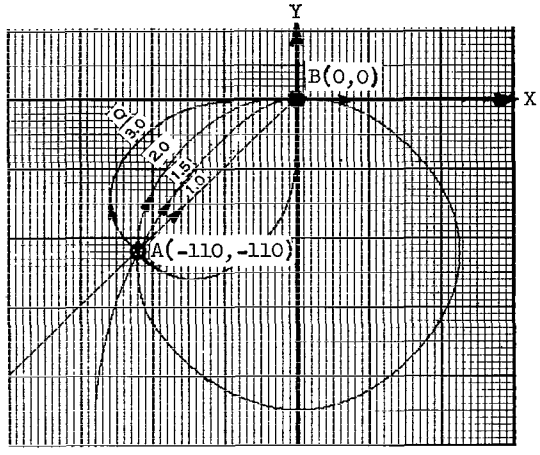



Figure 46.- Q-family connections from A to B.

## CONCLUDING REMARKS

The prime objective of this research was to evolve a simple controller for automatically generating a smooth curve through an ordered set of data points even for unequal arguments, closed contours, and raw data. The "figure of merit" for the generated curve was to be based on how well it resembled one properly drawn with french curves. Tacitly assumed was that sufficient input data were given to represent faithfully the curve without resorting to contextual information, such as characteristics of the data

source (e.g., frequency content, bandwidths, or intentional discontinuities), or recognizing that the input data fall into some analytic function or geometric shape. The effect of context was forcibly demonstrated when the figure eight was redrawn in the context of two circles; the resulting curve was significantly less distorted. Therefore, for the figure eight the sufficient input data assumption was violated. The foregoing and other experimental curves do indicate that given sufficient input data, the controller generates curves of equal fidelity to the skilled draftsman using french curves. Another way of stating this fact is: if enough input data are given so that the same curve results when drawn (out of context) by several skilled draftsmen, then the controller will also produce the same curve. The word "same" does not require a quantitative definition because even a highly subjective interpretation is perfectly adequate if consistently applied to both draftsmen and controller.

Considering the magnitude of the task which it performs, the controller satisfies also the secondary objectives of simplicity and speed.

Langley Research Center,
National Aeronautics and Space Administration,
Langley Station, Hampton, Va., April 6, 1966.

## APPENDIX

## DERIVATION OF STEERING ANGLE $\phi$

In figure 1 is a set of input data points a, b, c, d, e through which a smooth curve is drawn. Assume that the first portion of the curve is transformed to the new (primed) coordinate system of figure 2 such that the new origin is centered at point b and that the positive X'-axis is alined with $\beta$, the slope of the curve at b. This special case is analyzed in the more convenient transformed system and then generalized by means of the transformation itself. This procedure is particularly attractive since most of the parameters of the problem are angles which are invariant under translations and are simply rotated under coordinate rotation. These properties can be seen in figure 47 where under translation (from system 1 to system 2) the line pb does not rotate and the X-axis reference direction does not change; hence $\psi$ is invariant. However, under coordinate rotation (from system 2 to system 3), the reference direction changes by the amount of the rotation; hence $\psi$ changes by an equal amount and, thus

$$\psi' = \psi - \beta \tag{33}$$

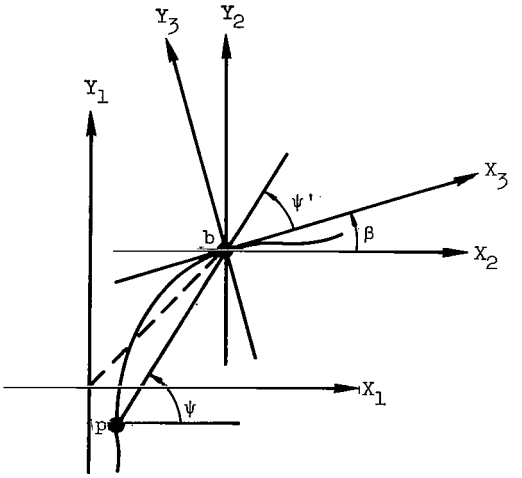



Figure 47.- Effect of transformation on $\psi$.

Another useful property is the invariance under rotation (and of course under translation) of a change in $\psi$. Thus, since $\beta$ is constant for $x_a \leq x_p < x_b$,

$$d\psi' = d\psi - d\beta = d\psi \tag{34}$$

Consider the present problem to be that of determining the angle $\phi'$ associated with the tangent of figure 2. The angle $\phi'$ can then be transformed to $\phi$ which can be used to guide point p from a to b.

Since b' is a destination of p', the direction of b' with respect to p' is apparently an important parameter. The angle $\psi'$ is therefore defined, as indicated in figure 2, as the instantaneous angle of the line connecting b' and p' as p' moves along the curve – that is,

$$\psi' = \tan^{-1}\left(\frac{y'}{x'}\right)$$

where x' and y' are the coordinates of p'.

## APPENDIX

Let $\phi' = \phi'(\psi')$ and, in the interest of obtaining simple relationships, assume that (eq. (1))

$$\phi' = Q\psi'$$

where $Q$ is a constant. Several significant cases then result.

Case I – $Q = 1$: Obviously case I is the linear case in which

$$\phi' = \psi' = \tan^{-1}\left(\frac{y'}{x'}\right) = c$$

where $c$ is a constant for $x'_a \leq x' < x'_b$. Hence $\phi'$ would guide $p'$ over a constant slope or linear path from $a'$ to $b'$.

Case II – $Q = 2$: From equation (1),

$$\phi' = 2\psi'$$

In figure 2 it is seen that $\frac{dy}{dx} = \tan \phi'$. Therefore,

$$\frac{dy}{dx} = \tan 2\psi' = \frac{2 \tan \psi'}{1 - \tan^2\psi'} = \frac{2\left(\frac{y}{x}\right)}{1 - \left(\frac{y}{x}\right)^2} = f\left(\frac{y}{x}\right) \tag{35}$$

a first order, homogeneous differential equation. (Prime with $x$ and $y$ is implicit.) Let

$$v = \frac{y}{x} \tag{36}$$

then $y = vx$ and

$$\frac{dy}{dx} = v + x\frac{dv}{dx} = \frac{2v}{1 - v^2}$$

Separating the variables gives

$$\frac{x}{dx} = \frac{\frac{2v}{1 - v^2} - v}{dv} = \frac{2v - v + v^3}{dv(1 - v)^2}$$

Therefore,

$$\frac{dx}{x} = \frac{\left(1 - v^2\right)dv}{v\left(v^2 + 1\right)} \tag{37}$$

By use of partial fractions,

$$\frac{1 - v^2}{v\left(v^2 + 1\right)} = \frac{A}{v} + \frac{B}{v^2 + 1} \tag{38}$$

and therefore,

$$A = \left.\frac{1 - v^2}{v^2 + 1}\right|_{v=0} = \frac{1}{1} = 1$$

Substituting $A = 1$ in equation (38) gives

$$\frac{1 - v^2}{v\left(v^2 + 1\right)} = \frac{1}{v} + \frac{B}{v^2 + 1} = \frac{v^2 + 1 + Bv}{v\left(v^2 + 1\right)}$$

Hence,

$$B = -2v$$

Substituting the values of $A$ and $B$ into equation (38) and equation (38) into equation (37) results in

$$\frac{dx}{x} = \left(\frac{1}{v} - \frac{2v}{v^2 + 1}\right)dv$$

Integrating gives

$$\ln|x| = \ln|v| - \ln|v^2 + 1| + \ln c$$

and, hence,

$$\ln\left[\frac{|x|}{|v|}\left(v^2 + 1\right)\right] = \ln c$$

Taking the antilogarithm gives

$$\pm c = x\left(\frac{1}{v} + v\right)$$

Therefore, from equation (36),

$$\pm c = x\left(\frac{x^2 + y^2}{xy}\right) \tag{39}$$

By substituting $2r$ for $c$,

$$x^2 + y^2 \pm 2yr = 0$$

and thus

$$x^2 + (y \pm r)^2 = r^2$$

Therefore, the resulting path for $\phi' = 2\psi'$ is one of a family of circles or radii r, centered on the Y-axis and displaced from the origin in both positive and negative directions by a distance equal to their radii. (See fig. 48.)

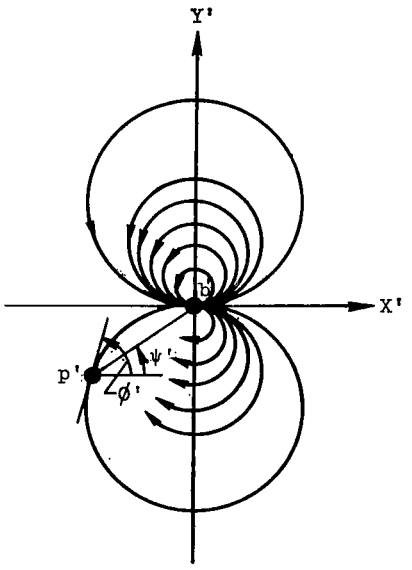



Figure 48.- Circular paths for $\Phi' = 2\psi'$.

## APPENDIX

Case III – Q = 3/2: From equation (1),

$$\phi' = \frac{3}{2}\psi'$$

Polar coordinates will be used to show that case III also yields a familiar geometric curve. First, however, it is convenient to introduce the lemma

$$\tan \delta = r \frac{d\theta}{dr}$$

where $r$ and $\theta$ are the standard polar variables and $\delta$ is the angle between the radius vector and the tangent line (fig. 49).

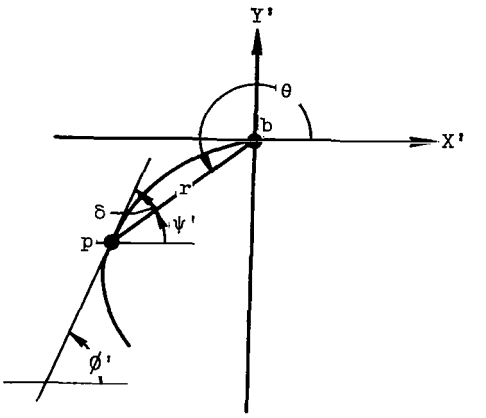



Figure 49.- Relation of $\delta$ to $\Phi'$, $\psi'$, and $\theta$.

For the proof of the lemma, assume that the curve is given in the form $r = r(\theta)$, where $r(\theta)$ is differentiable. From figure 49, it is seen that

$$\tan \delta = \tan(\phi' - \psi')$$

$$= \tan(\phi' - \theta + \pi)$$

Therefore,

$$\tan \delta = \tan(\phi' - \theta)$$

$$= \frac{\tan \phi' - \tan \theta}{1 + \tan \phi' \tan \theta}$$

$$= \frac{\frac{dy}{dx} - \frac{y}{x}}{1 + \frac{dy}{dx}\left(\frac{y}{x}\right)}$$

$$= \frac{x\,dy - y\,dx}{x\,dx + y\,dy} \tag{40}$$

Recall from the polar to Cartesian transformation that

$$x = r \cos \theta \tag{41a}$$

$$y = r \sin \theta \tag{41b}$$

and, therefore,

$$dx = -r \sin \theta\, d\theta + \cos \theta\, dr \tag{41c}$$

$$dy = r \cos \theta\, d\theta + \sin \theta\, dr \tag{41d}$$

Substituting equations (41) into equation (40) gives

$$\tan\delta = \frac{r\cos\theta(r\cos\theta\,d\theta + \sin\theta\,dr) - r\sin\theta(-r\sin\theta\,d\theta + \cos\theta\,dr)}{r\cos\theta(-r\sin\theta\,d\theta + \cos\theta\,dr) + r\sin\theta(r\cos\theta\,d\theta + \sin\theta\,dr)}$$

$$= \frac{r^2\cos^2\theta\,d\theta + r^2\sin^2\theta\,d\theta}{r\cos^2\theta\,dr + r\sin^2\theta\,dr} = \frac{r^2 d\theta}{r\,dr} = r\frac{d\theta}{dr}$$

which proves the lemma. From this lemma and figure 49,

$$r\frac{d\theta}{dr} = \tan\delta = \tan(\psi'/2) = \tan\left(\frac{\theta - \pi}{2}\right) = -\cot(\theta/2)$$

Separating the variables results in

$$\frac{dr}{r} = -\frac{d\theta}{\cot(\theta/2)} = -\frac{\sin(\theta/2)d\theta}{\cos(\theta/2)} \qquad (42)$$

Since the numerator is the differential of the denominator (except for a factor of 1/2 on the right-hand side) equation (42) is easily integrated to give

$$\ln r = 2\ln\cos(\theta/2) + \ln 2a$$

where $r > 0$, $\cos(\theta/2) > 0$, $a > 0$, and $\ln 2a$ is the integration constant. Taking the antilogarithm gives

$$r = 2a\cos^2(\theta/2) \qquad (-\pi < \theta < \pi)$$

and, therefore,

$$r = a(1 + \cos\theta) \qquad (43)$$

the classical form of the cardioid of figure 50.

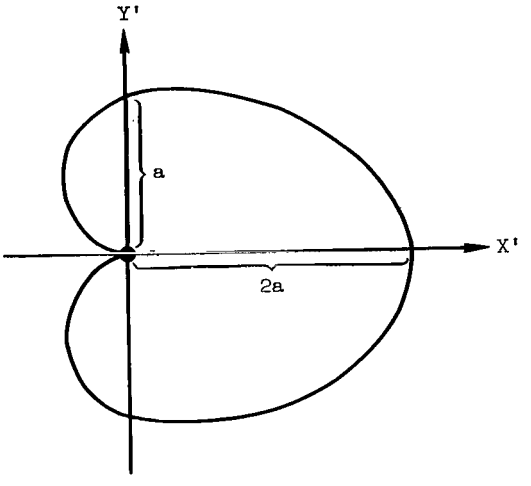



Figure 50.- Cardioid for $\Phi' = \frac{3}{2}\psi'$.

Case IV – Q = 3: From equation (1),

$$\phi' = 3\psi'$$

Again, from the lemma and figure 49,

$$r\frac{d\theta}{dr} = \tan\delta = \tan(2(\theta - \pi)) = \tan 2\theta$$

Separating the variables results in

$$\frac{dr}{r} = \frac{d\theta}{\tan 2\theta} = \frac{(\cos 2\theta)d\theta}{\sin 2\theta} \qquad (44)$$

which is easily integrated to give

$$\ln r = \frac{1}{2}\ln(\sin 2\theta) + \ln a$$

where $r > 0$, $\sin 2\theta > 0$, $a > 0$ (fig. 51), and $\ln a$ is the integration constant. Taking the antilogarithm yields

$$r^2 = a^2 \sin 2\theta \qquad (45)$$

the classical form of the lemniscate (ref. 13) of figure 51.

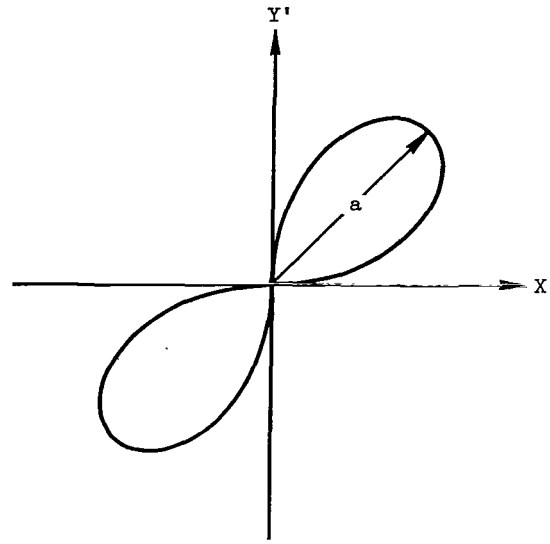



Figure 51.- Lemniscate for $\Phi' = 3\psi'$.

Since most of the equations of the foregoing figures are recognizable more easily in polar coordinates, a polar generalization, which includes cases II to IV (case I is considered to be trivial), is now derived. The relations

$$\delta = \phi' - \psi'$$

and

$$\psi' = \theta - \pi$$

from figure 49 with equation (1)

$$\phi' = Q\psi'$$

give

$$\delta = (Q - 1)(\theta - \pi)$$

Therefore the lemma can be expressed as

$$r\frac{d\theta}{dr} = \tan\left[(Q - 1)(\theta - \pi) + \pi k\right]$$

where $k$ is any integer. Separating the variables results in

$$\frac{dr}{r} = \frac{\cos\left[(Q - 1)(\theta - \pi) + \pi k\right]d\theta}{\sin\left[(Q - 1)(\theta - \pi) + \pi k\right]}$$

which is easily integrated to give

$$\ln r = \frac{1}{Q - 1}\ln \sin\left[(Q - 1)(\theta - \pi) + \pi k\right] + \frac{1}{Q - 1}\ln c$$

where $r > 0$, $\sin\left[(Q - 1)(\theta - \pi) + \pi k\right] > 0$, $c > 0$, and $\frac{1}{Q - 1}\ln c$ is the integration constant. Taking the antilogarithm yields the polar generalization

## APPENDIX

$$r^{Q-1} = c \sin\left[(Q - 1)(\theta - \pi) + \pi k\right] \qquad \left(\pi\left(1 - \frac{k}{Q - 1}\right) < \theta < \pi\left(1 - \frac{k - 1}{Q - 1}\right)\right) \quad (46)$$

where the domains of $\theta$ result from the conditions that

$$\left(\sin\left[(Q - 1)(\theta - \pi) + \pi k\right] > 0\right) \Rightarrow \left(0 + 2\pi m < \left[(Q - 1)(\theta - \pi) + \pi k\right] < \pi + 2\pi m\right)$$

for any integer m, and $2\pi m \subset k\pi$.

For case II, in which $Q = 2$, equation (46) becomes for $k = 1$

$$r = c \sin \theta \qquad (0 < \theta < \pi)$$

and for $k = 0$

$$r = -c \sin \theta \qquad (\pi < \theta < 2\pi)$$

which agrees with equation (39), as can be seen from the following transformation to Cartesian coordinates:

$$\pm c = \frac{r}{\sin \theta} = \frac{r^2}{r \sin \theta} = \frac{x^2 + y^2}{y}$$

For case III, in which $Q = 3/2$, equation (46) becomes for $k = 1$

$$r^{1/2} = c \sin\left(\frac{\theta}{2} + \frac{\pi}{2}\right) = c \cos\left(\frac{\theta}{2}\right) \qquad (-\pi < \theta < \pi)$$

Squaring both sides gives

$$r = c^2 \cos^2\left(\frac{\theta}{2}\right) = c^2 \frac{1}{2}(1 + \cos \theta)$$

which agrees with equation (43), where $\frac{c^2}{2} = a$.

For case IV, in which $Q = 3$, equation (46) becomes for $k = 2$ and $k = 0$, respectively,

$$r^2 = c \sin 2\theta \qquad \left(0 < \theta < \frac{\pi}{2}, \quad \pi < \theta < \frac{3\pi}{2}\right)$$

which agrees with equation (45), where $c = a^2$. Also for $k = 1$ and $k = -1$, respectively,

$$r^2 = -c \sin 2\theta \qquad \left(\frac{\pi}{2} < \theta < \pi, \quad \frac{3\pi}{2} < \theta < 2\pi\right)$$

Thus, equation (46) becomes

$$r^{Q-1} = c \sin\left[(Q - 1)\theta\right] \tag{47}$$

for integer values of $Q > 0$ and for $k = Q - 1$, and

$$\begin{aligned} r^{2(Q-1)} &= c^2 \sin^2\left[(Q - 1)\theta + \frac{\pi}{2}\right] \\ &= c^2 \cos^2\left[(Q - 1)\theta\right] \\ &= c' \left\{1 + \cos\left[2(Q - 1)\theta\right]\right\} \end{aligned} \tag{48}$$

for integer values of $(Q + 0.5) > 1$ and for $k = Q - 0.5$ and $c' = \frac{c^2}{2}$.

Other popular geometric shapes can probably be found by solving equation (47) for additional integer values of $Q$ or solving equation (48) for additional "half-integer" values. Also, $Q$ may be quartered or further subdivided to obtain other shapes. Without attempting to exhaust all of the important shapes, however, a significant conclusion is stated in the following theorem which is based on figure 49:

Theorem: If $\theta$ is the polar angle of a point on a curve and $\phi$ is the tangent angle of the curve at the point, then for $-\pi \leqq (\psi = \theta - \pi) < \pi$ simple linear relationships exist between $\phi$ and $\psi$, such that $p$ generates smooth well-known geometric shapes which go through the coordinate origin at zero slope.

All that remains to be proved is the origin zero-slope portion. That the origin is a solution of equation (46) is easily seen by substitution; its slope may be determined as follows:

$$\lim_{p \to 0} r^{Q-1} = 0 = \lim_{p \to 0} c \sin\left[(Q - 1)(\theta - \pi) + \pi k\right]$$

where $p \to 0$ means "as point $p$ approaches the origin." Therefore,

$$\lim_{p \to 0} \left[(Q - 1)(\theta - \pi) + \pi k\right] = \pi m$$

where $m$ is any integer. Letting $m = k$ gives

$$\lim_{p \to 0} \left[(Q - 1)(\theta - \pi)\right] = 0$$

and since $(Q - 1) \neq 0$

$$\lim_{p \to 0} \theta = \pi$$

Therefore, the curve goes through the origin at zero slope.

## REFERENCES


1. Mergler, Harry Winston: A Numerical Machine Tool Control System Operating From Coded Punched-Paper Tape. Publ. No. 20,943, Univ. Microfilms, Inc., 1956.
2. Henegar, H. B.: New Continuous Path System Uses DDA Interpolator. Control Eng., vol. 8, no. 1, Jan. 1961, pp. 71-76.
3. Kaiwa, Toshimasa; and Inaba, Seiuemon: Latest Japanese Numerical Control Features. Control Eng., vol. 8, no. 10, 1961, pp. 88-91.
4. Batte, William Granville: A Digital Higher Order Interpolation Path Controller. Ph.D. Thesis, Case Inst. Technol., 1965.
5. Ninke, William H.: A Two-Dimensional Absolute Digital Data Path Control System. Rept. No. EDC-1-63-18 (NsG-36-60), Digital Systems Lab., Case Inst. Technol., Sept. 1963.
6. [Mergler, H. W.]: Notes on Digital Control Systems Engineering. Vols. 1 and 2, Eng. Div. of Case Inst. Technol.
7. Goldberg, Samuel: Introduction to Difference Equations. Sci. Eds., Inc., 1961.
8. Kaplan, Wilfred: Advanced Calculus. Addison-Wesley Press, Inc., 1952.
9. Ledley, Robert Steven: Programming and Utilizing Digital Computers. McGraw-Hill Book Co., Inc., c.1962, pp. 127-132.
10. Lee, Thad; and Mix, Gerhart: TRICE Lecture Notes. CSP-156, Packard Bell Computer, Sept. 1963.
11. Horton, William P.: Shift Counters. 3C 013-6, Computer Control Co., Inc.
12. Phister, Montgomery, Jr.: Logical Design of Digital Computers. John Wiley & Sons, Inc., c.1958.
13. Burington, Richard Stevens: Handbook of Mathematical Tables and Formulas. Fourth ed., McGraw-Hill Book Co., c.1965.

*"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."*

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958.

# NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Technical information generated in connection with a NASA contract or grant and released under NASA auspices.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

TECHNICAL REPRINTS: Information derived from NASA activities and initially published in the form of journal articles.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities but not necessarily reporting the results of individual NASA-programmed scientific efforts. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

***Details on the availability of these publications may be obtained from:***

**SCIENTIFIC AND TECHNICAL INFORMATION DIVISION**

**NATIONAL AERONAUTICS AND SPACE ADMINISTRATION**

**Washington, D.C. 20546**